Richer Object Representations for Object Class Detection in Challenging Real World Images

Thesis for obtaining the title of
Doctor of Engineering Science
(Dr.-Ing.)
of the Faculty of Natural Science and Technology
of Saarland University

by
Bojan Pepik, Dipl. Eng.

Saarbrücken
December, 2015

Day of Colloquium	21th of December, 2015
Dean of the Faculty	Univ.-Prof. Dr. Markus Bläser

Examination Committee

Chair	Prof. Dr. Antonio Krüger
Reviewer, Advisor	Prof. Dr. Bernt Schiele
Reviewer	Prof. Dr. Christian Theobalt
Reviewer	Prof. Silvio Savarese
Academic Assistant	Dr. Yusuke Sugano
Date of submission	25th of November, 2015

ABSTRACT

Object class detection in real world images has been a synonym for object localization for the longest time. State-of-the-art detection methods, inspired by renowned detection benchmarks, typically target in-image localization of objects. At the same time, due to the rapid technological and scientific advances, high-level vision applications, aiming at understanding the visual world as a whole, are coming into the focus. The diversity of the visual world challenges these applications in terms of representational complexity, robust inference and training data. As objects play a central role in any vision system, it has been argued that richer object representations, providing higher level of detail than modern detection methods, are a promising direction towards understanding visual scenes. Besides bridging the gap between object class detection and high-level tasks, richer object representations also lead to more natural object descriptions, bringing computer vision closer to human perception. Inspired by these prospects, this thesis explores four different directions towards richer object representations.

First, we design 3D object representations, providing natural and compact descriptions of 3D object shape and geometry. Driven by the three-dimensional nature of objects, we gradually build a suite of 3D representations, capturing global 3D properties like viewpoint and coarse geometry, but also local object properties like 3D volumetric parts and detailed 3D shape. In an extensive evaluation on challenging benchmarks, we demonstrate excellent recognition performance of the 3D representations in 2D images, achieving comparable performance to state-of-the-art object detection methods.

Second, we show that fine-grained representations can be successfully utilized in 3D scene understanding and object class detection. Fine-grained information yields strong 3D geometric constraints, e.g. metric size of objects, which we further exploit for 3D scene understanding tasks. In addition, we verify that fine-grained representations can further boost object class detection, even when facing scarce training data for fine-grained categories.

Third, we demonstrate that occlusion-aware object representations can aid object class detection in driving scenarios. Building on non-randomness of occlusions, we explore contextual information around the occluded object, aiming at representing the characteristic occluder-occludee patterns. We confirm the benefits of the occlusion-aware representation in terms of improved detection performance on occluded objects and also overall.

And fourth, we delve deeper into understanding state-of-the-art convolutional neural net representations from the perspective of object class detection. By dissecting the performance across different appearance factors, we analyze what current state-of-the-art architectures have learned, and in a second step we illustrate what can these architectures actually learn.

In summary, this thesis presents encouraging findings in different dimensions towards richer object representations, illustrating that richer object representations can facilitate high-level applications, providing richer, more detailed and natural object descriptions. In addition, the presented representations attain high performance rates, at least on par or often superior to state-of-the-art methods.

ZUSAMMENFASSUNG

Detektion von Objektklassen in natürlichen Bildern war lange Zeit gleichbedeutend mit Lokalisierung von Objekten. Von anerkannten Detektions-Benchmarks inspirierte Detektionsmethoden, die auf dem neuesten Stand der Forschung sind, zielen üblicherweise auf die Lokalisierung von Objekten im Bild. Gleichzeitig werden durch den schnellen technologischen und wissenschaftlichen Fortschritt abstraktere Bildverarbeitungsanwendungen, die ein Verständnis der visuellen Welt als Ganzes anstreben, immer interessanter. Die Diversität der visuellen Welt ist eine Herausforderung für diese Anwendungen hinsichtlich der Komplexität der Darstellung, robuster Inferenz und Trainingsdaten. Da Objekte eine zentrale Rolle in jedem Visionssystem spielen, wurde argumentiert, dass reichhaltige Objektrepräsentationen, die höhere Detailgenauigkeit als gegenwärtige Detektionsmethoden bieten, ein vielversprechender Schritt zum Verständnis visueller Szenen sind. Reichhaltige Objektrepräsentationen schlagen eine Brücke zwischen der Detektion von Objektklassen und abstrakteren Aufgabenstellungen, und sie führen auch zu natürlicheren Objektbeschreibungen, wodurch sie die Bildverarbeitung der menschlichen Wahrnehmung weiter annöhern. Aufgrund dieser Perspektiven erforscht die vorliegende Arbeit vier verschiedene Herangehensweisen zu reichhaltigeren Objektrepräsentationen.

Erstens entwerfen wir 3D-Objektrepräsentationen, die natürliche und kompakte Beschreibungen von 3D-Objektform und Geometrie zur Verfügung stellen. Bedingt durch die dreidimensionale Eigenheit von Objekten erstellen wir schrittweise eine Folge von 3D-Repräsentationen und erfassen globale 3D Eigenschaften wie Blickwinkel und grobe Geometrie, aber auch lokale Objekteigenschaften wie 3D-volumetrische Teile und detaillierter 3D-Umriss. In einer ausführlichen Evaluation auf anspruchsvollen Benchmarks zeigen wir eine hervorragende Erkennungsleistung der 3D-Repräsentationen in 2D-Bildern und erreichen eine Leistung, die Objektdetektionsmethoden auf dem neuesten Stand der Forschung vergleichbar ist.

Zweitens zeigen wir, dass detailgenaue Repräsentationen beim 3D-Szenenverständnis und bei der Objekt-klassendetektion erfolgreich eingesetzt werden können. Detailgenaue Informationen liefern starke 3D-geometrische Nebenbedingungen, wie zum Beispiel die metrische Größe von Objekten, was wir für Aufgaben des 3D-Szenenverständnisses weiter nutzen. Des Weiteren belegen wir, dass detailgenaue Repräsentationen die Objektklassendetektion weiter voranbringen können, sogar bei einer begrenzten Anzahl von Trainingsdaten für detaillierte Kategorien.

Drittens legen wir dar, dass Objektrepräsentationen, die Verdeckung berücksichtigen, bei der Detektion von Objektklassen in Straßenszenen hilfreich sein können. Aufbauend auf der Nicht-Zufälligkeit von Verdeckungen erforschen wir kontextbezogene Informationen bezüglich des verdeckten Objekts und versuchen, charakteristische Verdeckungsmuster zu repräsentieren. Wir bestätigen die Vorzüge der Repräsentation, die Verdeckung berücksichtigt, sowohl hinsichtlich einer verbesserten

Detektionsleistung bei verdeckten Objekten als auch im allgemeinen.

Viertens befassen wir uns eingehender mit dem Verständnis neuester „convolutional neural net representations“ (Repräsentationen faltender neuronaler Netze) unter dem Gesichtspunkt der Objektklassendetektion. Durch getrennte Betrachtung der Leistung für verschiedene Faktoren des Erscheinungsbildes analysieren wir, was gegenwärtige moderne Architekturen gelernt haben, und in einem zweiten Schritt zeigen wir auf, was diese Architekturen tatsächlich lernen können.

Zusammengefasst präsentiert diese Arbeit in verschiedener Hinsicht ermutigende Ergebnisse auf dem Weg zu reichhaltigeren Objektrepräsentationen und zeigt, dass reichhaltigere Objektrepräsentationen abstrakte Anwendungen erleichtern können, indem sie reichhaltigere, detailliertere und natürliche Objektbeschreibungen liefern. Darüber hinaus erreichen die vorgestellten Repräsentationen hohe Leistungsraten und sind damit anderen Methoden auf dem neuesten Stand der Forschung mindestens gleichwertig oder oft sogar überlegen.

ACKNOWLEDGEMENTS

First and foremost, I want to thank my supervisor, Prof. Bernt Schiele for giving me the opportunity to be part of his lab. His enormous energy has been a constant source of motivation and inspiration to push forward. His persistent and valuable advice have allowed me to rise from a total novice in computer vision to a qualified researcher in the field. Furthermore, I would like to express my gratitude to my closest collaborators, Michael Stark and Peter Gehler. Michael has been the closest person I could rely on and without his dedication and energy, our work would not have been even halfway to where it is now. I would like to thank Peter as well, for his original views and thoughts, always providing refreshing new ideas and directions. I am truly grateful to Prof. Silvio Savarese for serving as an external reviewer on the thesis committee. I would also like to thank Prof. Christian Theobalt for being part of the thesis committee.

I am grateful to my other collaborators who I had chance to work with. Special thanks for Rodrigo Benenson for his fruitful discussions and unconditional help. I am grateful to Tobias Ritschel for bringing the world of Graphics closer to me.

I would like to express my sincere gratitude to all members of the CVMC team for creating an excellent and warm working environment, but also for sharing all the fun and happy moments we had together. In particular, I thank my former office mate, Leonid Pischulin for his positive attitude and unique sense of humor, Mohamed Omran for being a great friend and Siyu Tang for her unique and cheerful attitude. Life in the CVMC group would have been a lot more difficult if it wasn't for Connie Balzert and her excellent organizational work. Special thanks to our administrators, Marcus Rohrbach and Jan Hosang, for addressing our technical issues.

I would like to thank Gaurav Sharma, Yusuke Sugano, Shanshan Zhang, Seong Joon Oh, Mateusz Malinowski and Walon Wei-Chen Chiu for proofreading this thesis.

I am grateful to all my friends in Saarbrücken who helped me retain a healthy work-live balance: Milivoj Simeonovski, Kiril Panev, Marinela Spasova, Dragan Milcevski, Evica Ilieva, Maximilian Dylla, Mohamed Yahya, Christina Tefloudi, Jasmina Bogoeska and Levi Valgerts. Special thanks to Kiril Panev for his insightful analysis of sports, especially football, to Milivoj Simeonovski for sharing his positive attitude to life, Maximilian Dylla for spreading his incredible lust for partying and Dragan Milchevski for being the most knowledgable person I have ever met.

My sincerest thanks to all my friends in Macedonia, who always encouraged me in pursuing my dreams, before and after coming to Germany. I owe particular thanks to my best friend and best man Dejan Skrceski for his support and charisma.

Lastly, I would like to thank my family for their unconditional support. Most of all, I would like to thank my wife Monika for always believing in me and constantly providing her encouragement and understanding.

CONTENTS

1 INTRODUCTION

Contents

One of the most impressive and remarkable human abilities is to parse and understand visual scenes. Humans, on a daily basis identify people, recognize objects, estimate object properties like colors, shapes, distances, predict actions and perform various other cognitive activities, all within parts of a second. The outstanding quality of human perception has inspired scientists to aim at creating machine vision systems able to perceive the world at least as good as humans, if not even better. The impact of such systems on everyday life would be immense and very broad, ranging from applications like autonomous driving, via specialized industrial tasks to household applications like cleaning and cooking. However, due to the high complexity of understanding visual scenes entirely and despite best efforts and rapid technological developments, machine vision is still far from reaching the quality of human perception. To address the sheer complexity of the problem, scientists have adopted a "divide and conquer" strategy and rather than addressing the grand problem of machine vision directly, research has focused on solving well defined, and very specialized vision tasks like object categorization (Krizhevsky *et al.* (2012)), object class detection (Girshick *et al.* (2014)), pedestrian detection (Benenson *et al.* (2013)), object tracking (Wu and Nevatia (2007); Tang *et al.* (2015)), human pose estimation (Yang *et al.* (2012)), image segmentation (Noh *et al.* (2015)), scene layout estimation (Liu *et al.* (2015)), etc.

As objects play a key role in human perception, among all specialized vision tasks, object categorization and object class detection represent core technologies for any vision system. The key pillar towards successful categorization and detection are powerful object representations that can be robustly matched to image evidence. Indeed, a lot of progress has been made in this area. Back in the early days of computer vision, object recognition methods strived towards building rich and expressive

Figure 1.1: Complex outdoor (Geiger *et al.*, 2012; Lin *et al.*, 2014) and indoor (Nathan Silberman and Fergus, 2012; Xiao *et al.*, 2013) visual scenes.

three-dimensional object representations (Marr and Nishihara, 1978; Brooks, 1981; Pentland, 1986; Lowe, 1987) that were subsequently matched to images of simplistic scenes with objects. Although executed in highly controlled environment, these methods firmly acknowledged the inherent 3-dimensional nature of objects (Hoiem and Savarese, 2011). More recently, with the arrival of large scale object recognition and detection datasets (Everingham *et al.*, 2007; Deng *et al.*, 2009; Geiger *et al.*, 2012), robust object recognition methods addressing challenging real world scenarios have begun to emerge. Relying either on a combination of powerful machine learning techniques e.g. SVM (Cortes and Vapnik, 1995), boosting (Freund and Schapire, 1997), random forests (Breiman, 2001) and robust pre-computed image representations e.g. SIFT (Lowe, 2004), HOG (Dalal and Triggs, 2005), ShapeContext (Belongie *et al.*, 2000) or on deep end-to-end learning frameworks (Atlas *et al.*, 1988; Krizhevsky *et al.*, 2012), modern object recognition methods have delivered excellent performance, successfully confronting the challenges of real world images, like cluttered scenes, partial objects (Felzenszwalb *et al.*, 2010; Girshick *et al.*, 2014; Wang *et al.*, 2013) and high intra-class variability. However, while early vision research focused on rich and highly descriptive object representations, modern methods are typically limited in that respect, as they typically target image localization and categorization only, neglecting the need for detailed and interpretable representations.

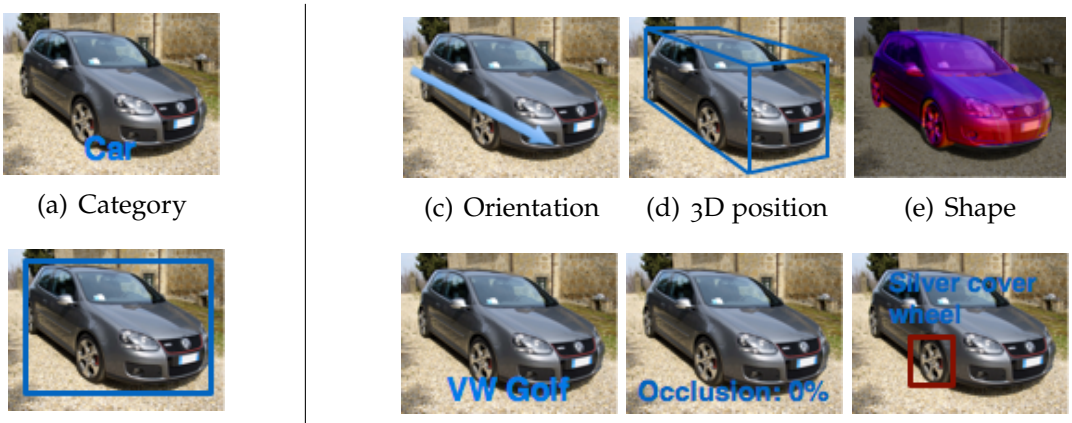
As we are reaching excellent performance for specialized vision tasks like object class detection (Wang *et al.*, 2013; Girshick *et al.*, 2014), more challenging high-level applications, like 3D scene understanding (Geiger *et al.*, 2014; Schwing *et al.*, 2013)

and autonomous driving (Gehrig and Stein, 1999) are getting into the focus. Aiming at understanding the scene as a whole, these high-level tasks are challenged by the high complexity of the visual world. As illustrated in Figure 1.1, the visual world, both outdoor and indoor, contains information on different levels of granularity: scene-level information capturing scene properties like scene layout, topology and geometry (Geiger *et al.*, 2011); sub-scene level representing groups of interacting objects like walking people (Tang *et al.*, 2014), parked cars Xiang *et al.* (2015b); Wu *et al.* (2015a); and object-level information, describing the position, orientation and various other object properties. Given this myriad of visual information, highly descriptive scene representations (Geiger *et al.*, 2014; Kim *et al.*, 2013; Wang *et al.*, 2015) explicitly representing the three granularity levels and their interplay, have been dominating the field (Mottaghi *et al.*, 2014; Yao *et al.*, 2012). In particular, reminiscent of the early days of computer vision, rich geometric object representations (Zia *et al.*, 2013a; Xiang *et al.*, 2015b) in combination with strong contextual and geometric constraints have been considered key to success (Wojek *et al.*, 2011; Bao and Savarese, 2011; Gupta *et al.*, 2011). Evidently, there is a gap between the rich and detailed representation required for visual scene understanding, and the limited object representation delivered by state-of-the-art object categorization and detection methods. As a result, current high-level applications are limited to coarse-grained object representations, where reasoning is typically constrained to the level of entire objects (Geiger *et al.*, 2014; Wojek *et al.*, 2010, 2013).

As object representations are the core technology for any vision system, it becomes apparent that more detailed object representations, providing natural, geometric and contextual object information are going to be a valuable asset (Xiang *et al.*, 2015b; Aubry *et al.*, 2014) for high-level vision tasks. In addition, the added level of detail would bring object recognition closer to human vision, as people are able to infer a diverse set of object properties. As a consequence, inspired by the current excellent advances in object recognition, motivated by the far reaching potential of high-level vision applications, driven by the ideal of building systems that can reach human vision, in this thesis we aim at bridging the *representational gap* between scene representations for the emerging high-level vision tasks on the one hand, and object representations in the state-of-the-art recognition methods on the other hand. In particular, we investigate building *richer* object representations than the current state-of-the-art representations, providing natural, geometric and contextual object descriptions, that can further facilitate understanding of visual scenes. At the same time, we aim at object representations that can be reliably matched to image evidence, attaining the high performance rates of current state-of-the-art methods.

1.1 RICHER OBJECT REPRESENTATIONS FOR OBJECT CLASS DETECTION

The central question this thesis revolves around is illustrated in Figure 1.2. Given an image, current object class detection methods target in-image object localization,



(a) Category (c) Orientation (d) 3D position (e) Shape

(b) 2D Bounding box (f) Subordinate category (g) Occlusion level (h) Attributes and parts

Figure 1.2: (Left) Typical object description by modern object representations. (Right) Higher level of detail provided by richer object representations.

providing axis-aligned 2D bounding box around the object, as shown in Figure 1.2(b), along with the object category. And indeed, the object descriptions provided by such representations can be sufficient for many applications, like image retrieval or object recognition in web collections. Obviously, looking at Figure 1.2, humans can infer many more details, e.g. that the car has dark gray color, we are looking at the frontal right side of it, it is not too far from the camera, it looks like a VW Golf, it is fully visible and has silver covers for the wheels. This diverse and structured set of object descriptions requires object representations that can faithfully capture the variation of object appearance across different appearance factors like viewpoint, shape and context but at the same time, representations that are discriminative enough in distinguishing objects from different categories and from an arbitrary background. This thesis explores several different directions towards obtaining *richer* object representations in the context of several applications: *(i) 3D object representations* for 3D object recognition and more broadly 3D scene understanding, *(ii) fine-grained representations* for scene understanding and object class detection, *(iii)* explicit *occlusion representations* for driving scenarios and finally, *(iv)* understanding *convnet representations* and their behavior w.r.t. various appearance factors. In the following, we go in further detail about each of them.

First, we focus on 3D object representations. Many practical applications would greatly benefit from reasoning about 3D object properties like viewpoints, 3D parts and shapes. Consider for example the scenes illustrated in the top row of Figure 1.1. From the orientation of the moving vehicles we can infer the structure of the road and the directions of the moving traffic. Similarly, the orientation of the parked vehicles can inform the driver about potential crossing pedestrians. In addition to contextual constraints, object viewpoint poses functional constraints as well. For example, as can be seen in the last two rows of Figure 1.1, the viewpoint of a chair defines the seating position of a person, the viewpoint of the TV set constraints the furniture

and the plausible viewing directions. In a robotic scenario, the viewpoint of a cup poses grasping constraints (Jiang *et al.*, 2012) for the grasping hand. Obviously, object viewpoint is an important cue for scene understanding (Geiger *et al.*, 2011; Wojek *et al.*, 2011), but also for object recognition (Gu and Ren, 2010; Tulsiani and Malik, 2015), as the appearance of objects varies drastically across viewpoints. This has led to the creation of several viewpoint datasets, like 3D Object Classes (Savarese and Fei-Fei, 2007) and ICARO (Lopez-Sastre *et al.*, 2010) providing coarse viewpoint information, and also more recently the KITTI (Geiger *et al.*, 2014) and the Pascal3D+ (Xiang *et al.*, 2014a) datasets that come with continuous viewpoint labels in challenging real world scenarios. Inspired by the evident importance of object viewpoint for vision applications, *viewpoint representations* play a central role in this thesis. We thoroughly explore several discrete (chapters 3 and 6) and continuous (chapters 4 and 6) viewpoint representations in both controlled and realistic scenarios.

While viewpoint affects object appearance globally, object parts are crucial in representing local appearance deviations. In fact, object parts provide strong discriminative cues towards object categorization and detection (Marr and Nishihara, 1978; Felzenszwalb and Huttenlocher, 2005). One of the main characteristics of parts is that they are volumetric (consider the car wheel in Figure 1.2) and are shared across different viewpoints of the same object. Motivated by the widely accepted importance of object parts, compact and natural part representations are the second crucial pillar in this thesis. In particular, in chapters 3-5 we explore *3D part representations* that are shared across viewpoints, allowing for part correspondences across different views of the same object. In Chapter 6 we introduce a representation of 3D keypoints, which are the driving force behind a 3D object class detection method.

Among all object properties, 3D shape represents the most detailed, high-resolution description of object geometry. As such, 3D object shape promises to aid detailed 3D scene-level reasoning (Zia *et al.*, 2014a), in challenging realistic scenarios (Xiang *et al.*, 2015b), even when multiple objects jointly interact in the scene (Zia *et al.*, 2014b). At the same time, free online repositories of 3D CAD models, like Trimble 3D Warehouse[1] (2.5M models) and Turbosquid[2] (300K models), provide high-resolution geometric descriptions for many categories. Inspired by the far reaching potential of 3D object geometry, scalable *3D shape representations* and 3D data sources in particular, constitute the third important pillar in this thesis. Throughout the thesis we rely on 3D data sources to represent 3D object geometry, and in Chapter 6 specifically we explore 3D CAD alignment to objects in images towards obtaining robust 3D object detection in challenging real world scenarios.

While 3D object representations help tasks like autonomous driving and 3D scene understanding, fine-grained information can also yield strong 3D geometric constraints that can be useful for high-level tasks. Therefore, in a second direction, we explore using *fine-grained representations* from the perspective of 3D scene understanding. Subordinate object information imposes strong geometric constraints in terms of object sizes (Chapter 7) that can be further exploited for various scene under-

[1] 3dwarehouse.sketchup.com
[2] www.turbosquid.com

standing tasks like depth estimation. Going even further, object class detection can greatly benefit from fine-grained object representations. For example, a bus is more easily confused with an average car model, rather than with a highly specialized car-type model. Inspired by this fact and challenged by the scarce fine-grained data distribution for many specialized subordinate classes, in Chapter 8 we introduce detailed, multi-view, fine-grained object representations learned from extremely sparse fine-grained data across viewpoints, relying on robust knowledge transfer techniques.

Occlusion is one of the main sources of failure for any vision system. Occluded and partially visible objects are a frequent sight in urban walking as well as driving areas. As object evidence is reduced with partially visible objects, detecting partial objects is an extremely challenging task. Motivated by the idea that occlusions are not just random noise but rather a signal that can be explicitly represented, in a third direction, we explore *occlusion-aware representations*. Focusing on an autonomous driving scenario, we observe that occlusions are indeed not random, and in fact happen in specific and well defined patterns. By explicitly representing the appearance of the occluded and the occluder object in these patterns, we establish a very powerful occlusion-aware object representation in Chapter 9.

Recently, tremendous advances have been made in object categorization and class detection by employing deep end-to-end convolutional neural network (convnet) based methods (Krizhevsky *et al.*, 2012; Girshick *et al.*, 2014). Convnets have shown remarkable potential in providing robust separation of objects from background. Although very powerful, convnet-based object representations are not thoroughly understood. Therefore, in a fourth direction, we aim at providing deeper *understanding of convnet representations*. Deemed to be invariant to various appearance factors like viewpoint and size, in Chapter 10 we challenge this perception and aim at analyzing what state-of-the-art convnet architectures actually learned. Furthermore, we continue analyzing what current convnet architectures can learn when provided with more training data.

The added expressiveness of richer object representations should not come at the cost of sacrificing object detection performance. Typical counter-example are the powerful 3D shape representations from the past (Malik, 1987; Kanade, 1980; Brooks, 1981; Pentland, 1986; Lowe, 1987; Zia *et al.*, 2013a). Although very rich, these 3D object representations have proven extremely difficult to match to 2D image evidence. Matching such representations to images has proven difficult and inefficient even nowadays (Zia *et al.*, 2015). Therefore, in this thesis we aim at representations that are highly descriptive and rich, but at the same time retain the high detection rates and efficient inference of state-of-the-art methods.

In addition to the expectation of high detection rates, in this thesis we assume single RGB images as input. This requirement has two important implications. First, it makes the task significantly more difficult compared to having depth data as well (Gupta *et al.*, 2014; Nathan Silberman and Fergus, 2012) where the depth cues allow for easier estimation of 3D shapes and object occlusion interactions, or compared to a multi-view setting (Yebes *et al.*, 2015). On the other hand, as a second

implication, having a single image as input, makes our method generally applicable to a wide variety of unconstrained real world scenarios. At the same time, we do not impose any assumption regarding object appearance, whether it is indoor or outdoor, or about the background. Therefore, the object representations in this thesis are general and are meant to be applied in unconstrained real world scenarios, or in other words, in the wild.

To conclude, many practical applications in vision and robotics require and would greatly benefit from object representations providing additional information other than image location of objects. To that end, in this thesis we contribute richer object class representations to facilitate various vision applications. We explore 3D object representations (Chapters 3, 4, 5, 6), fine-grained representations (Chapters 7, 8), occlusion representations (Chapter 9) and understanding convnet representations (Chapter 10).

1.2 CHALLENGES TOWARDS RICHER OBJECT REPRESENTATIONS

Figure 1.1 illustrates the complexity of indoor and outdoor visual scenes. At the same time, it hints at the difficulties and challenges of object class detection. Clutter, partial objects, varying illumination, viewpoints and sizes are only a few of the many pitfalls object representations are confronted with. In this section we emphasize and discuss these challenges in greater detail. First, we explain the challenges arising in 3D object representations, then we dive into the challenges of object class detection in general and in the end we discuss the specific challenges fine-grained representations are facing.

1.2.1 Challenges towards 3D object representations

In this section, we start with discussing the challenges specific to 3D object representations in the context of 3D object recognition and scene understanding.

Descriptive and discriminative 3D representations Standard object class detection strives for invariant and discriminative object representations (Krizhevsky *et al.*, 2012; Girshick *et al.*, 2014). 3D object recognition, however, aims at representations that are discriminative, but at the same time descriptive enough allowing reliable estimation of 3D object properties like viewpoint and 3D shape. Therefore, instead of being invariant, the ideal 3D object representation should vary according to the factors of interest like viewpoints, 3D parts and shapes.

3D data source 3D data can come from different sources, starting from reconstruction techniques, like structure from motion (SfM, Glasner *et al.* (2012)), using depth sensors like Kinect (Shotton *et al.*, 2013) or Lidar scanners (Geiger *et al.*, 2012), to using 3D computer aided design (CAD) models (Stark *et al.*, 2010). There are two major challenges regarding 3D data. First of all, the domain

shift between 3D data and 2D images imposes a strong requirement to rely on machine learning methods capable of transferring useful information across domains. A second challenge is to use these different sources alone or in conjunction in order to obtain the most useful 3D object representation. The three different data sources have their respective pros and cons. While SfM and depth sensors can be easily registered to image data, they typically come with label noise and result in coarser representations (Kar *et al.*, 2015). CAD models on the other hand, provide very rich, detailed and accurate 3D information, however, they typically lack realism (Stark *et al.*, 2010; Liebelt and Schmid, 2010).

Matching 3D representations to 2D evidence With object representations capturing detailed object shape and part information in 3D, test time inference aims at aligning these representations to image evidence. Due to the ambiguities arising when matching 3D models to 2D evidence (Yoruk and Vidal, 2013; Zia *et al.*, 2013a; Brooks, 1981; Pentland, 1986; Lowe, 1987) these models typically could not be reliably applied in challenging real world scenarios. In fact, there seems to be a natural trade-off between having rich object representations on the one hand, and robust model matching to images on the other hand.

Viewpoint representation Humans are proficient in providing coarse viewpoint predictions. For example, people can easily tell if they are looking straight into someones face, profile or back part of the head. While this coarse categorical viewpoint representation is sufficient for humans, there are quite a few applications like object grasping in robotics, where angular-accurate viewpoint representations are of great value. Either way, obtaining a descriptive viewpoint representation is challenging due to three reasons. First, the viewpoint labeling process, especially in the angular-accurate case, is extremely tedious and erroneous. Second, the viewpoint representation should be sensitive and descriptive to small angular viewpoint variations. And third, highly accurate viewpoint inference in large scale setups is challenging, in particular for discrete viewpoint representations, as one has to perform inference in the cross-product of object categories and viewpoints.

Multi-task learning Traditional object detection typically boils down to a 1-vs-all binary classification task (Girshick *et al.*, 2014; Felzenszwalb *et al.*, 2010). While this has proven to be sufficient for object class detection, in 3D object recognition we are interested in solving several tasks simultaneously, including viewpoint prediction, 3D shape estimation, localizing object parts in 3D, estimating distance etc. Thus the multi-task learning algorithm has to be able to cope with the prerequisites of the different tasks, which might be contradictory at times.

3D object detection evaluation metric Establishing a proper 3D object detection metric is difficult due to several factors. First, standard object detection datasets come without 3D ground truth annotations. Second, the metric should jointly capture several factors at hand: 3D position, orientation and shape. Third, as

objects are more distant from the camera methods get more prone to errors, as small errors in pixel space result in large errors in 3D space.

1.2.2 Challenges in object class detection in general

Object class detection is a highly researched area and it's challenges have been widely emphasized in the object detection literature. As object class representations coping with realistic scenarios are paramount of this thesis, in the following we outline these challenges.

Invariant object representations One of the main challenges in object class detection and object recognition in general is to learn representations that are invariant (Quiroga *et al.*, 2005) to appearance variations of a specific object category. A good object class representation is deemed to be insensitive to appearance differences across different instances of the same class, due to factors like viewpoint, size, shape, context. As the same time, the representation should be discriminative enough to allow separation of the object class of interest from random background and other object classes.

Understanding object representations In recent years, object representations learned directly from data based on deep neural networks have shown outstanding recognition performance Krizhevsky *et al.* (2012), and have surpassed previous highly engineered representations (HOG Dalal and Triggs (2005), SIFT Lowe (2004)). Although very powerful and discriminative, these deep neural net representations have not been well understood, in terms of how they internally handle intra-class variability and inter-class discrimination. This has inspired speculative comparisons of deep object representations to the human cognitive system on different levels. For example, researchers Gupta *et al.* (2014) have been investigating the existence of so called "grandmother neurons" (Barlow, 1972) - hypothetical neurons that respond only to specific objects and concepts (e.g. grandmothers).

Object occlusion & truncation Objects of interest can often be only partially visible, either due to other objects (occlusion), or due to contact with the image boundary (truncation). Partial objects have proven difficult to detect, due to the smaller image support, the high diversity of occlusion scenarios and also due to being typically underrepresented in object detection benchmarks.

Small objects The size of objects in images is directly correlated with the distance of objects to the camera. While objects closer to the camera provide higher level of detail, objects far away from the camera result in smaller image support. In the latter case, object class detection has to infer the presence of an object at a particular location from less information, resulting in more difficult object discrimination.

Object localization Objects tend to vary significantly w.r.t. aspect ratio and size, due to factors like shape, distance, viewpoint, articulation and interaction with other objects. This large set of factors make the precise and tight localization of objects in images challenging. At the same time, quite a few applications like pose estimation and 3D registration require precise localization of not only objects, but also object parts. However, standard detection methods cast the representation learning as a 1-vs-all classification problem, explicitly ignoring the localization quality.

Object parts detection Given that object localization is a challenging problem, localizing and detecting object parts is even more difficult. Object parts and specific keypoints on the surface of the object are at the core of many computer vision applications like pose estimation, fine-grained recognition, 3D registration and fundamental matrix estimation. In fact, part based object detection has been one of the dominant paradigms in computer vision (Felzenszwalb and Huttenlocher, 2005). Parts tend to be small, occluded and articulated. This reason and the fact that many applications require high part localization accuracy, render part localization a difficult and challenging problem.

Large and representative data Finally, recent success in computer vision and machine learning has been driven by discriminative learning methods applied to large training corpora. While large datasets are common in image recognition (Russakovsky *et al.*, 2014; Lin *et al.*, 2014), object class detection has been lagging behind due to the more expensive labeling process. This results in unbalanced and heavy-tailed data distributions across appearance factors, which represent serious challenge for any object class detection algorithm.

1.2.3 Challenges in fine-grained object representations

Last, we explore the challenges in fine-grained object representations, from the perspective of higher-level applications.

Expert knowledge for data acquisition People are particularly proficient when it comes to recognizing *base* level categories. When we see an object, we immediately know whether it's car, airplane, sheep, human etc. However, when it comes to recognizing categories on a deeper level of the object hierarchy, for example recognizing an aircraft *model* or *manufacturer* (Maji *et al.*, 2013), car *types* (Stark *et al.*, 2012), bird *species* (Welinder *et al.*, 2010a), a non-expert typically struggles to recognize the categories correctly. This becomes more apparent as categories become more detailed or more fine-grained. As only experts in the area are competent to provide accurate fine-grained categorization, collecting large and detailed collections for fine-grained recognition has proven expensive and hard.

Heavy-tailed data distributions In our daily lives we encounter different object categories with different frequency. We encounter cars and people on a daily

basis. At the same time, we see categories like kangaroos maybe once per year. This observation is even stronger for fine-grained categories. Real world datasets naturally reflect this observation and typically (Salakhutdinov *et al.*, 2011) follow a heavy-tailed distribution across categories, with only a few classes containing lots of training data and most classes with only a few training examples. Therefore, fine-grained recognition methods are challenged with learning both descriptive and discriminative representations from only a few training examples. This issue becomes even more prominent when jointly tackling multi-view and fine-grained object representations.

Learning fine-grained representations Learning fine-grained representations is faced with two major challenges. First, in order to disambiguate among fine-grained categories it has to focus on very detailed, local information (e.g. *black footed albatross*). As this information is usually latent, fine-grained recognition methods have to automatically discover discriminative category information which might not be even visible all images. Second, as the number of classes is large and at the same time data is scarce, fine-grained learning methods have to be scalable, efficient and be able to reuse information across categories.

Leveraging fine-grained object representations Finally, fine-grained representations are useful for many high-level vision applications. The added level of detail by fine-grained representations results in additional constraints (e.g. in terms of object size) that can be exploited. However, task specific challenges arise, e.g. leveraging fine-grained representations for multi-view recognition confronts the problem of extremely sparse data across viewpoints for specific fine-grained categories (Mottaghi *et al.*, 2015).

1.3 THESIS CONTRIBUTIONS

After discussing the challenges, in this section we summarize the contributions of this thesis. We follow the same route, first we explore our contributions towards 3D object representations, then we continue with discussing contributions to object class detection in general and fine-grained representations. While this section groups the contributions by topic, for a chapter-wise summary we refer the reader to section 1.4.

1.3.1 Contributions to 3D object representations

Descriptive and discriminative 3D representations We enrich object representations step-by-step, gradually introducing 3D information in our models. Additionally, we require that the models attain high detection performance. In chapters 3 and 5 we first contribute a set of multi-view object detectors inspired by the deformable parts model (DPM, Felzenszwalb *et al.* (2010)) reaching state-of-the-art performance simultaneously for object localization and viewpoint estimation performance on several outdoor benchmarks, and interestingly, are still com-

petitive with deep learning-based methods. We proceed with establishing a 3D deformable parts based model in chapters 4 and 5, parameterizing objects as well as parts fully in 3D space. Finally, in chapter 6 we contribute a full 3D object detection pipeline estimating the 3D position, orientation and shape by aligning CAD models to objects. Inspired by deep learning methods, the 3D object detection pipeline achieves state-of-the-art 3D object detection performance.

3D data source Going from viewpoint to full 3D object representations, throughout the thesis we use CAD models as the main source of 3D geometric information. In chapters 3, 4, 5 we show that CAD models can be effectively used to reliably learn about category-level 3D geometry, even when observed through simplistic wire-frame renderings. Going beyond these simplified renderings, in chapter 10 we compare different CAD rendering techniques with varying levels of realism, with the goal of improving convnet-based detection methods. In addition to using this 3D data source for training data enhancement, we also leverage CAD data to estimate object geometry, and in chapter 6 we illustrate that exemplar-based CAD collections can be robustly matched to 2D images for several object categories (12 categories in Pascal3D+) relying only on a few object keypoints to drive the alignment. Lastly, in chapters 5 and 6 we illustrate that CAD models can be successfully combined with real data, with the real world data being responsible for the learning of realistic appearance models and the CAD models contributing to the learning of the 3D object geometry and shape.

Matching 3D representations to 2D evidence We develop efficient matching techniques for all 3D object representations explored in this thesis. In chapters 4, 5, we match the part-based 3D object representation by instantiating viewpoint-specific representations in arbitrary views. In chapter 6 we contribute an efficient CAD-to-image alignment method.

Viewpoint representations We explore discrete versus continuous viewpoint representations at two different occasions. First, in chapters 4, 5 in the context of 3D deformable part models and second, in the context of convnet representations in chapter 6. We experimentally verify that continuous viewpoint representations are not only more compact, scalable and more natural than the discrete ones, but also consistently result in better viewpoint estimation, leading to state-of-the-art performance on different datasets. Apart from the scalable continuous viewpoint representations presented in chapters 4 and 6, in chapter 4 we also contribute a greedy coarse-to-fine viewpoint estimation strategy which results in major speed-ups, while preserving the performance of the exhaustive inference techniques.

Multi-task learning The multi-view object detection methods in chapters 3 and 5 rely on a joint object localization and viewpoint estimation training framework. Approaching this multi-task setup as structured output prediction problem, we

construct a loss function jointly capturing the two tasks at hand, achieving excellent joint localization and viewpoint estimation performance on challenging outdoor driving scenarios like KITTI (Geiger *et al.*, 2012) but also on general object detection benchmarks like Pascal3D+ (Xiang *et al.*, 2014a). The joint task learning framework leads to significantly fewer opposite view confusions, which is the main reason for the excellent viewpoint estimation performance. Chapter 6 on the other hand, investigates sequential training for different tasks, starting from object class localization via viewpoint estimation and keypoint localization to 2D-3D registration.

3D object detection evaluation metric As evaluating 3D object hypotheses is challenging, in chapter 6 we propose to approach the problem via two proxy tasks: first via joint object localization and viewpoint estimation, evaluating the localization and viewpoint estimation quality and second, via segmentation accuracy addressing all three goals together: location, orientation and shape estimation. In addition, previous joint object localization and viewpoint estimation metrics typically discretize the viewpoint (Xiang *et al.*, 2014a), resulting in discretization dependent evaluation. As this is sub-optimal, in chapter 6 we contribute a discretization independent evaluation metric, aiming to capture the whole spectrum of viewpoint errors.

1.3.2 Contributions to object class detection in general

Invariant object representations In chapter 10, focusing on the question "what did convnets learn", we explore invariances of three state-of-the-art convnet architectures: AlexNet (Krizhevsky *et al.*, 2012), GoogleNet (Szegedy *et al.*, 2014a) and VGG16 (Simonyan and Zisserman, 2015). We experimentally show that these three architectures are not invariant to many appearance factors like viewpoints, object size, truncations and partial occlusions and in fact have similar weak points. Interestingly, we observe strong correlation between these weak points and several confounding factors: the underlying data statistics and the image support of an object.

Understanding object representations In addition to having common weaknesses, in chapter 10 we illustrate that there is a high discrepancy in how convnets handle common cases vs outliers. With significantly worse performance on small, heavily truncated and occluded objects, we conclude that the overall performance can not be simply improved by generating more outlier data. In fact, using synthetic data generation techniques, we illustrate that convnet performance can only be boosted by sampling data from the common cases. Furthermore, by exploiting the complementarity among the three architectures, we achieve state-of-the-art performance on a highly relevant object detection benchmark.

Small objects Having understood that detecting small objects is a major challenge

for convnets, we contribute two techniques (chapter 10) to improve detection on small objects and outliers in general. First, we explore data augmentation techniques and second, we investigate whether training specialized models for different object sizes can boost performance. We find that simply generating data is not enough in the case of small objects, and either architectural changes are needed and/or adequate learning methods.

Occlusion & truncation Driven by the idea that occlusions are not random, but rather heavily depend on the context the objects appear in, in chapter 9 we develop occlusion-aware object class detection methods. In particular, we first automatically discover characteristic object-object occlusion patterns in the data capturing typical occlusion cases. By learning specialized representations for occlusion patterns, we achieve excellent detection performance both on the occlusion cases, and also overall. Our occlusion-aware detectors achieved, at the time of publication, state-of-the-art detection performance on a renowned autonomous driving dataset and won a top tier outdoor object detection challenge (RMRC, Urtasun *et al.* (2013)).

Object localization In chapters 3 and 5 we contribute a structured output learning framework, jointly addressing object class recognition on the one hand, and bounding box driven object localization, on the other hand. By explicitly encoding the object localization performance in the loss function of the structured max-margin learning method (Yu and Joachims, 2009), the localization aware object detection consistently outperforms the standard one-vs-all max-margin learning framework on the Pascal VOC 2007 dataset (Everingham *et al.*, 2007).

Object parts detection As object part detection is a difficult problem, it requires powerful detection techniques. To that end, in chapter 6 we contribute an R-CNN (Girshick *et al.*, 2014) inspired detector for specific object keypoints. The proposed detector achieves state-of-the-art part detection performance on Pascal3D+ (Xiang *et al.*, 2014a) dataset. Furthermore, in chapters 3, 4 and 5 we introduce models that represent object parts in 3D space effectively allowing the model to establish part correspondences across viewpoints. We show that the model can reliably localize corresponding parts across different views of the same object, even when the views have and ultra-wide baseline.

Large and representative data Even though wireframe like renderings of CAD models are not realistic in appearance, we show in chapters 3, 4, 5 that this type of training data can still boost the detection performance when used in combination with real world data. In chapter 10, we investigate using different rendering techniques with varying levels of realism and conclude that indeed realism plays a crucial role in boosting performance.

1.3.3 Contributions to fine-grained representations

Fine-grained data acquisition We contribute two fine-grained recognition datasets. First, in chapter 7, we introduce a data set of car types with 1904 internet images annotated with 14 different categories and viewpoint annotations at 5° angular resolution. The dataset comes with a rich set of labels, including bounding boxes, viewpoints, car types and metric sizes. As internet car images tend to be fully visible and unrealistic in appearance, in chapter 8 we contribute fine-grained annotations on a realistic driving outdoor scenario containing partial and small objects. We contribute detailed car *brand*, *type* and *sub-type* annotations on the KITTI tracking dataset (Geiger *et al.*, 2012).

Heavy-tailed data distributions Still, these newly acquired datasets still suffer from the problem of scarce training data (see chapter 8) for many fine-grained categories. This issue is even more obvious when considering multi-view and fine-grained object representations together. To address that problem, we contribute a hierarchical knowledge transfer technique (chapter 8), transferring knowledge in two different directions. First, from *base* level categories to subordinate categories and second across viewpoints. The proposed knowledge transfer techniques allow to learn rich multi-view representations for different levels of the object hierarchy, even when the given class of interest has only a few viewpoints represented in the dataset. Furthermore, in chapter 8 we empirically confirm that the learned fine-grained representations can be leveraged for better object (*base* level) recognition performance.

Learning fine-grained representations In addition to the knowledge transfer technique presented in chapter 8, in chapter 7 we contribute a joint, large scale, fine-grained and multi-class learning technique. The joint learning technique successfully disambiguates fine-grained categories and reaches state-of-the-art performance on the newly introduced dataset.

Leveraging fine-grained object representations After having properly trained fine-grained representations, we can leverage them in various applications. First, in chapter 7 we contribute a 3D object localization technique relying on the metric information provided by the fine-grained labels. Second, in chapter 8 we successfully illustrate that fine-grained representations result in better object detection performance achieving state-of-the-art performance on the newly introduced fine-grained KITTI dataset (Geiger *et al.*, 2012).

1.4 THESIS OUTLINE

This section gives an overview of the thesis. It provides thematical and chronological ordering of the chapters, and relates them to the original publications.

Chapter 2: Related Work In this chapter we provide an overview of prior work on

richer object representations. First, we focus on object class detection and prior work addressing general challenges in object recognition. We continue with related work towards richer object representations and in particular we focus on 3D, multi-view and fine-grained object representations.

Chapter 3: Teaching 3D Geometry to Deformable Part Models State-of-the-art object class detection methods deliver bounding box oriented object hypotheses. As pointed out earlier, there is an evident gap between the output of standard object class detection methods on the one hand, and high-level scene representations on the other hand. To that end, this chapter makes the first step in this thesis towards bridging this gap. First, it discusses multi-view object representations and second it focuses on 3D part parameterizations allowing for part correspondences across views. In addition, this chapter addresses joint object localization and viewpoint learning, approaching the two tasks in a multi-task, structured output learning framework.

The content of this chapter corresponds to the CVPR 2012 publication "Teaching 3D Geometry to Deformable Part Models" (Pepik *et al.*, 2012b).

Chapter 4: 3D Deformable Part Models While the previous chapter discusses multi-view representations, in this chapter we make a step forward and establish full 3D object representations based on the deformable parts (Felzenszwalb *et al.*, 2010) paradigm. The presented representations in this chapter are not only fully parameterized in 3D, which constitutes a more compact and natural object representation, but also allow for continuous appearance representations. In addition, this chapter addresses learning 3D representations in max-margin, structured output learning frameworks.

This chapter corresponds to the ECCV 2012 publication "3D^2*PM* - 3D Deformable Part Models" (Pepik *et al.*, 2012a).

Chapter 5: Multi-view and 3D Deformable Part Models While the previous two chapters independently present multi-view and 3D object representations, in this chapter we present a unified framework for deformable parts models. Viewing DPMs as conditional random fields, we present the models of the previous two chapters as specific instantiations of the deformable parts paradigm. In addition to the unified framework, in this chapter we provide broader and more extensive experimental evaluation, expanding on the number of object categories and detection benchmarks. This results in a more comprehensive comparison of the richer 3D object representations.

This content of this chapter corresponds to the PAMI 2015 publication: "Multi-view and 3D Deformable Parts Models" (Pepik *et al.*, 2015a).

Chapter 6: 3D Object Class Detection in the Wild Previous chapters discuss multi-view and 3D object representations consisting of a coarse set of 3D object parts. In this chapter, we make further advances towards even richer 3D object representations, allowing for detailed 3D shape representation and full camera

estimation. The 3D object detection method in this chapter uses convnet representations for object localization, keypoint detection and viewpoint estimation in a 3D detection pipeline aligning CAD models to objects in images. The presented method results in state-of-the-art 3D object detection performance on the Pascal3D+ dataset.

The content of this chapter correspond to the publication at the "3D from Single Image" workshop held at CVPR 2015: "3D Object Class Detection in the Wild".

Chapter 7: Fine-grained Categorization for 3D Scene Understanding. In contrast to previous chapters, in this chapter we focus on fine-grained object representations that can be further leveraged in the context of 3D scene understanding. In particular, we recognize that part geometry, appearance and their mutual interplay are powerful cues for fine-grained disambiguation. By building powerful multi-class categorization models, in a second step, we realize that fine-grained information comes with object metric information, practically for free in the case of rigid categories and we show that the added detail can be used for depth estimation.

The content of this chapter corresponds to the BMVC 2012 publication: "Fine-grained Categorization for 3D Scene Understanding" (Stark *et al.*, 2012). Bojan Pepik contributed with the best performing multi-class, part-based fine-grained recognition method, that was further used as a basis for the depth estimation experiments.

Chapter 8: Learning multi-view Priors from Sparse Viewpoint Data. While the previous chapter focused on fine-grained recognition for 3D scene understanding, in this chapter we use fine-grained information for the benefit of object class detection. In particular, we realize that fine-grained information would lead to reduced confusions at test time (e.g. a bus is more similar to a general car, rather than to a VW Golf) and aim at building rich multi-view representations at subordinate category level. However, fine-grained categories come with sparse viewpoint information, and thus this work investigates knowledge transfer across viewpoints, but also across levels in the object category hierarchy.

The content of this chapter is based on the ICLR 2014 publication: "Learning multi-view priors from Sparse Viewpoint Data" (Pepik *et al.*, 2014).

Chapter 9: Occlusion Patterns for Object Class Detection. Occluded objects are a major source of failure of many vision applications. Despite this fact, occlusions and the context surrounding an object can be a very useful cue in establishing better object class detection methods, as the contextual information is not random. To that end, in this chapter we approach occlusion as signal that should be modeled. In particular, we recognize that occlusions happen in specific and characteristic occlusion patterns. By automatically discovering the meaningful occlusion patterns in a dataset, we can then learn occlusion aware object class detectors, that not only capture the appearance of the occluded object, but also the appearance of the occluder.

The content of this chapter corresponds to the CVPR 2013 publication: "Occlusion Patterns for Object Class Detection" (Pepik *et al.*, 2013). Furthermore, our occlusion-aware object class detection method won the Reconstruction Meets Recognition Challenge (RMRC, Urtasun *et al.* (2013)).

Chapter 10: What is Holding Back Convnets for Detection? While in previous chapters we focused on learning richer object class representations for object class detection and recognition, in this chapter we take a different route and focus on understanding convnet image representations, again from the context of object class detection. In particular, we aim at understanding what did convnets learn and what they can learn. Furthermore, we question the big data paradigm by synthetically generating additional training data.

The content of this chapter corresponds to the GCPR 2015 publication "What is Holding Back Convnets for Detection?" (Pepik *et al.*, 2015b).

Chapter 11: Conclusions and Future Work. In this chapter we conclude the thesis and outline the disadvantages of the current contributions of the thesis. To that end, we further propose potential future directions overcoming the mentioned limitations. Furthermore, we give an outlook for richer object class representations and scene representations from a broader perspective, anticipating future research directions.

RELATED WORK 2

Contents

In this chapter, we revisit previous related work towards richer object representations. Due to the large volume of previous work in this area we focus only on related work that is both of high relevance to the field and tightly related to this thesis. To that end, in section 2.1 we first make a tour in object class detection in general, focusing on the ideas that dominated the field in the past. Then, in section 2.2 we give an overview of 3D object representations in the context of object recognition, detection and 3D scene understanding. Last, in Section 2.3 we review previous work on fine-grained representations.

2.1 OBJECT CLASS DETECTION

Object class detection is at the core of many computer vision problems, and as such has been addressed since the beginnings of computer vision. As objects are inherently three-dimensional, 3D object representations were the predominant paradigm back in the early days of computer vision. However, the difficulties arising from the ambiguities when matching 3D models to 2D evidence have diverted research towards more simplistic but robust 2D representations (Felzenszwalb *et al.*, 2010; Girshick *et al.*, 2011, 2014; Sermanet *et al.*, 2014; Wang *et al.*, 2013; Fidler *et al.*, 2013; Uijlings *et al.*, 2013). In particular, part-based representations (Felzenszwalb *et al.*, 2010; Andriluka *et al.*, 2009, 2012; Leibe *et al.*, 2008) relying on hand-engineered image features (Dalal and Triggs, 2005; Belongie *et al.*, 2000), dominated this field in the past. More recently, end-to-end methods (Krizhevsky *et al.*, 2012; Girshick

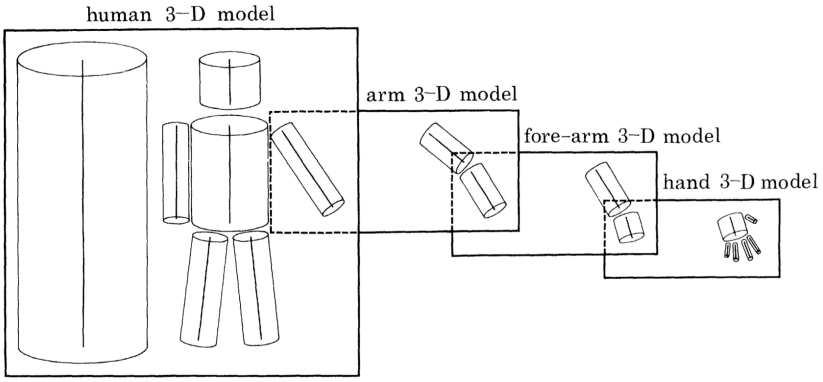


Figure 2.1: 3D object representation based on generalized cylinders. Figure from (Marr and Nishihara, 1978).

et al., 2014), directly learning representations from images in combination with object proposal methods (Uijlings *et al.*, 2013; Hosang *et al.*, 2014, 2015) have overtaken the lead, drastically changing the object detection landscape. Therefore, in this section we first review 3D object representations from the early days of computer vision, and then we provide and overview of both, part-based object representations and convnet inspired representations. In addition, we review previous work addressing detection of occluded objects.

2.1.1 Early vision

We start with a chronological overview of prominent modeling ideas from early days of computer vision. These works have served as an inspiration for most of the recent work on 3D object representations, which we review in section 2.2.

Nevatia and Binford (1977) is one of the first works to analyze images of complex curved objects obtained with laser range data. Aiming to represent the 3D shape of an object, since 3D shapes are invariant to many factors , this work adopts a part-based 3D object representation. Each object is segmented into geometric primitives called generalized cones. The segmentation results in a symbolic representation of object parts and symbolic connections defining the object topology. At recognition time, the object graph is matched against a database (memory) of stored exemplars with ground truth descriptions and matching is performed in a greedy bottom-up manner. The work is limited to instance level recognition of toy examples.

Marr and Nishihara (1978) study the representation of 3D shapes for the purpose of object recognition. The work provides a theoretical analysis of what are good criteria for examining 3D shape representations, and also the necessary requirements of a "good" object representation. The work concludes that a good 3D object representation should be modular in organization, allow for volumetric primitives and has to be defined in a object-centric coordinate system. The proposed representation in this work is similar to the one in Nevatia and Binford (1977). As illustrated in Figure 2.1, the object is modeled via generalized cylinders, using non-loopy object

topology, describing the relative part geometry via so called canonical axes. At test time, for a given image, first a canonical axis of the object is estimated, which is then, similar to Nevatia and Binford (1977), matched to a database of 3D exemplars, relying on shape descriptions invariant under projection.

Brooks (1981) introduced a general image understanding, model-based system called ACRONYM, allowing a user to specify the structure of the object category in an object graph and it's relations to other categories and sub-categories in a so called restriction graph. At test time, the specified 3D object representations, which also in this work consist of generalized 3D cylinders, are matched to image descriptions using a prediction graph. The system aims to interpret images by locating instances of modeled objects, following a coarse-to-fine matching procedure. At the same time, the work focuses on bottom-up evidence by predicting image features which enable identification of object instances. The final prediction graph consists of nodes which represent image features and edges which have to comply to geometric constraints imposed by the object model. The connected components in the graph represent the final object hypotheses. The work is limited to aligning 3D instances to images.

Lowe (1987) presents a 3D instance alignment method, that directly aligns a 3D model using the perceptual information from a 2D image. While most of the methods at the time relied on stereo or depth information, this work proposes a 3-stage procedure that works directly on images. First, a process of perceptual organization is used to search for groups and structures in the image that are invariant over a range of viewpoints. Second, it relies on a probabilistic ranking method to prune the search space for possible 3D model alignments and third, the 3D model is aligned to the object in an image via a process searching for spatial correspondences. The work fully relies on matching 3D line segments to computed 2D line segments in images and it works even in cluttered scenarios.

Dickinson *et al.* (1992) represent the object as a collection of object aspects. The method relies on 3D geometric primitives which are used to represent 3D objects in a database of CAD models resulting in a hierarchical aspect layout representation of objects. The aspect hierarchy consists of boundary groups, that are grouped into faces which are consequently grouped into objects aspects. While the aspects are defined in 2D, they are observed as projections of 3D volumetric primitives, which is the driving force for test time inference. To constrain the set of hierarchical relations, Dickinson *et al.* (1992) estimate conditional probabilities connecting different levels in the aspect hierarchy.

2.1.2 Part-based object representations

The explicit 3D representations from the early days of vision have proven hard to match to images. Therefore, part-based 2D representations, combining powerful machine learning techniques with robust image representations have dominated the field. Inspired by the seminal work of Fischler and Elschlager (1973), representing objects as a collection of parts connected with springs, researchers have intensively worked on part-based object representations in the past. Deemed to be robust to

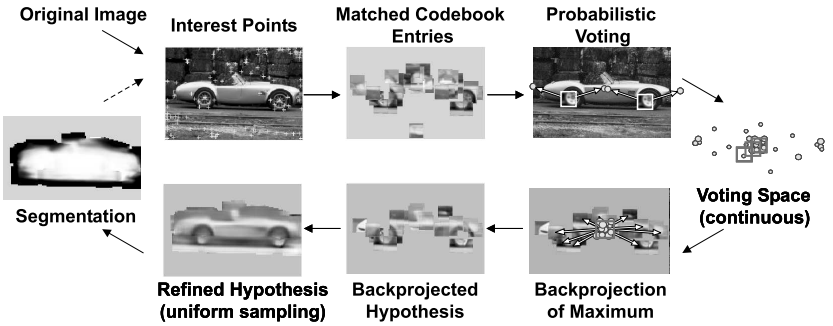


Figure 2.2: The implicit shape model (ISM). Figure from (Leibe *et al.*, 2004).

occlusions and not as data-hungry as deep neural nets, this type of object representation has been long favored in the literature. Therefore, in this section we review previous work in this area. In particular we focus on the constellation model, the implicit shape model, the deformable parts model and the pictorial structures model. The last two in particular, served as inspiration for several object representations in this thesis.

Implicit Shape Model. Considered conceptually simple and highly interpretable, the implicit shape model (ISM) has received increased attention in the past. Originally introduced by Leibe *et al.* (2004) for simultaneous object recognition and segmentation, the method has also been applied to a large variety of tasks like pedestrian detection (Seemann *et al.*, 2007; Wohlhart *et al.*, 2012), people tracking (Andriluka *et al.*, 2008) and 3D object recognition (Glasner *et al.*, 2011, 2012).

The ISM model presented in Leibe *et al.* (2004) has two major components (see Figure 2.2). The first component is the category-specific alphabet (codebook) of prototypical local patches, and the second, a spatial probability distribution specifying the relative codebook entry location, relative to the object center. At training time, each candidate training patch is first matched against the codebook and the relative position and scale for the activated codebook entry is stored. At test time, Leibe *et al.* (2004) rely on a generalized hough transform (Ballard, 1987), allowing the codebook entries to cast votes for candidate object hypotheses. Resulting in excellent recognition performance, the original ISM model has been further improved. In particular, Maji and Malik (2009) show that the performance can be significantly improved using a max-margin discriminative hough transform learning, assigning higher weights to codebook entries that allow for stronger background/foreground disambiguation. This resulted in excellent recognition performance on relatively simple datasets like ETHZ Shape Classes, UUIC Car and INRIA Horse. Following the same direction, Gall and Lempitsky (2009) introduce Hough forests, replacing the Hough transform with a discriminative random forest, directly mapping probabilistic votes given the candidate image patch, again illustrating that discriminative approaches boost the detection performance. Razavi *et al.* (2012) on the other hand, push the boundaries to a different level, showing outstanding results on Pascal

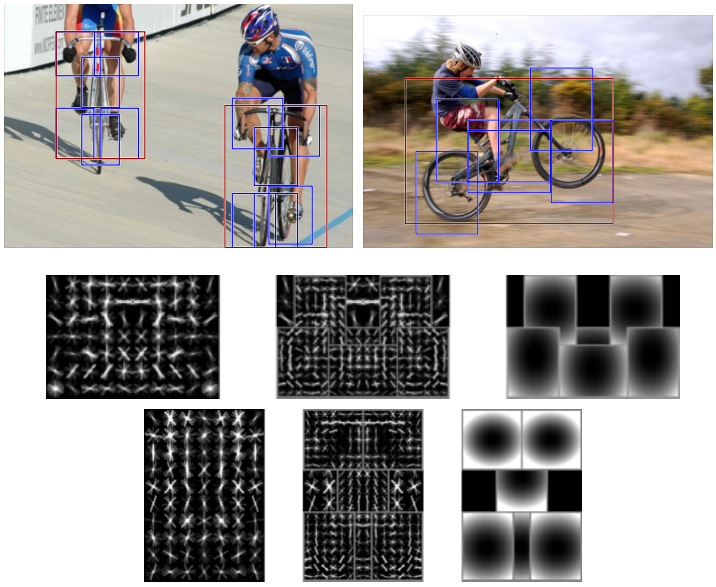


Figure 2.3: The deformable parts model (DPM). Figure from (Felzenszwalb *et al.*, 2010).

VOC 2007 datasets. Their latent Hough transform model augments the Hough transform with latent variables to enforce stronger spatial and scale consistency among the probabilistic votes. Wohlhart *et al.* (2012) revisit pedestrian detection and segmentation in the ISM voting framework and propose a graphical model on top of the probabilistic votes, efficiently inferring partially visible objects in an image, overcoming the need for non-maxima suppression as a post processing step.

Deformable Parts Model. Similar to the probabilistic interpretation of the spatial component in the ISM model that corresponds to a star-shaped part topology, the deformable parts model (DPM) also relies on a star-shaped constellation of parts, where each part is coupled and allowed to deform w.r.t. the object center. Introduced by Felzenszwalb *et al.* (2010), the DPM (see Figure 2.3) has been one of the major breakthroughs in object class detection due to it's excellent detection performance on challenging datasets (Pascal VOC). At training time, the DPM uses a discriminative max-margin learning framework, enabling the joint learning of latent object parts, totally circumventing the need for tedious part annotations. Furthermore, the star-shaped conditional random field with Gaussian pairwise potentials allows for efficient part inference via the generalized distance transform (Felzenszwalb and Huttenlocher, 2012) during test time. Typically the DPM is trained as a mixture model of several star-shaped components, where each component captures specific aspects (e.g. viewpoints) of the object class of interest. The excellent performance resulted in the DPM being rewarded the "Life-time Achievement" prize (Everingham *et al.*, 2010) by the Pascal VOC challenge.

Inspired by the tremendous advances by the original DPM, researchers have started addressing different aspects of the model. Ott and Everingham (2011) introduce shared parts across components to address run-time complexity, but also to

allow for sharing training data across components. Going a step further, Azizpour and Laptev (2012); Chen *et al.* (2014) show that part-level supervision can be beneficial for learning better models for animal categories, outperforming the original DPM model significantly on animal classes of the Pascal VOC dataset. Showing the benefits of strongly supervised DPMs, Azizpour and Laptev (2012) search for an optimal non-cyclic part topology, showing empirically that the Minimum Spanning Tree results is a well performing part topology in their case. In a next step, Chen *et al.* (2014) consider using a fully connected and flexible part-based model, explicitly addressing small objects, occluded parts and large deformations, ultimately leading to state-of-the-art performance on Pascal VOC. Apart from the model topology, researchers have also focused on building scalable deformable parts models. To that end, Pirsiavash and Ramanan (2012); Song *et al.* (2012) recognize that part convolutions represent a major bottleneck and therefore represent parts via a basis of filters, allowing for real-time DPM implementations. (Dubout and Fleuret, 2012) rely on the Fourier transform to speed-up filter convolutions in DPM. On the other hand, Dean *et al.* (2013) contribute a ultra scalable DPM approach by replacing the dot-product convolution with locality sensitive hashing. Last, Girshick *et al.* (2011) instead of phrasing the DPM learning as one-vs-all binary classification problem, it presents a max-margin structured output learning framework jointly optimizing for object localization and object recognition.

Pictorial Structures. Pictorial structures (Felzenszwalb and Huttenlocher, 2005) keep the tractability properties of the deformable parts model, while at the same time increasing the representational complexity of the model, modeling the object categories with a tree-like part topology. Inspired by the initial work of Fischler and Elschlager (1973) (see Figure 2.5), Felzenszwalb and Huttenlocher (2005) introduce a tree-like pictorial structures (PS) model combined with maximum-a-posteriori (MAP) inference based on the generalized distance transform.

The tree-like PS topology makes it especially suited for human detection and human pose estimation, where human parts like head, arms and torso are directly represented as unary potentials and the springs correspond to pairwise terms defining the permissible relative part configurations. To that end, Andriluka *et al.* (2009, 2011) revisit pictorial structures for human pose estimation, relying on discriminatively trained parts, combining AdaBoost (Freund and Schapire, 1997) with ShapeContext (Belongie *et al.*, 2000) features, and Gaussian pairwise terms to describe part relations. Achieving state-of-the-art performance on human pose estimation benchmarks, the work by Andriluka *et al.* (2009) inspired a lot of research towards richer and more powerful pictorial structures. Yang and Ramanan (2013) acknowledge the rigidity of the model and introduce flexible mixtures of parts for pose estimation, representing a part not with a single but multiple part templates. Following the same direction, Pishchulin *et al.* (2013a) use local mixtures of parts, and in addition introduce poselet conditioned pictorial structures, with the actual image content driving the choice of the specific part templates and pairwise terms that are actually used for a given image, leading to fast and accurate pictorial structures. In recent years, scientists also considered going beyond the tree-like model topology and

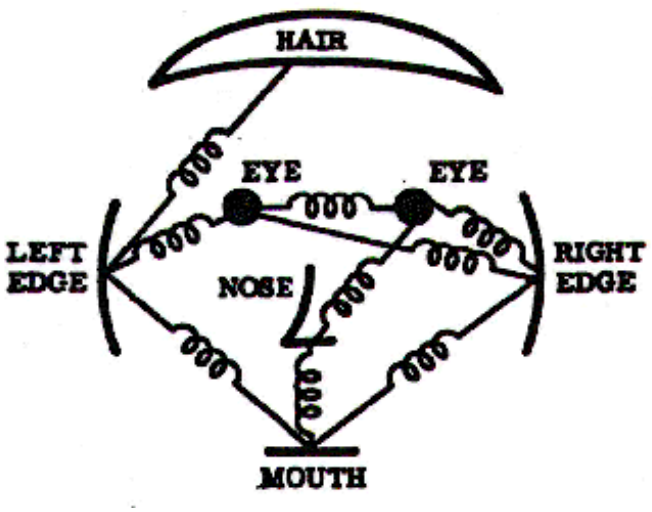


Figure 2.4: The idea behind pictorial structures (PS). Figure from (Fischler and Elschlager, 1973).

introduced loopy topologies. Sun *et al.* (2012b) use branch and bound optimization for efficient inference in a fully connected PS. Additionally, Kiefel and Gehler (2014) introduce fields of parts, a binary CRF model for human pose estimation. Following a binary parameterization of part positions, orientations and scales, the work presents a densely connected pictorial structures model combined with approximate part inference, resulting in superior performance compared to the local mixture of parts by Yang and Ramanan (2013).

Constellation Model. All previous part-based models employ a carefully designed model structure allowing for efficient inference, at the cost of having reduced model complexity. Contrary to that, the constellation model employs a fully connected graph structure, resulting in a powerful model of high complexity. Burl and Perona (1996) use the fully connected part-representation to detect planar objects. The plausible object deformations are represented through shape statistics, which is subsequently learned from examples. The method achieved very good face detection performance at the time of publication. In a follow up Burl *et al.* (1998), introduce the concept of "soft" part detectors, in an attempt to approximate an "optimal" object detector. At test time, the method resorts to a greedy heuristic for frontal face detection in images. Weber *et al.* (2000) avoid using strong part-level supervision, but introduce a constellation model with unsupervised part learning. Fergus *et al.* (2003) introduce a scale-invariant constellation model. Relying on salient image regions at different scales, this work models part appearance as Gaussian densities, learned via expectation maximization in maximum likelihood setting. The method showed promising results on faces, cars and animal categories. Stark *et al.* (2010) introduced discriminative learning of part appearance from synthetic data renderings. Relying on a set of CAD models of cars, this work learns viewpoint-specific part appearance terms, combining ShapeContext image descriptor and AdaBoost. This

work showed competitive detection performance on the 3D object classes dataset. More recently, Chen *et al.* (2014) employ a constellation model for animal detection. Relying on part annotations at training time, this work uses a DPM (Felzenszwalb *et al.*, 2010) inspired part learning. At test time, in order to reduce the computational burden, this work resorts to a large set of non-cyclic part topologies instead of the fully connected model. This work showed excellent animal detection performance on the Pascal VOC 2010 dataset.

2.1.3 Occlusion representations

Partial occlusions and partial objects in general, are one of the main sources of failures in many vision applications. Sensitivity to partial occlusion has so far mostly been considered a lack in robustness, with standard object detection methods treating occlusion as "noise rather than signal"[3]. While part-based representations implicitly enable reasoning about partial objects, still they also struggle with partially visible objects (Hoiem *et al.*, 2012). Therefore, addressing partial occlusion in real world imagery is an active research area. To that end, in this section we explore related work on occlusion representations for object class detection. We focus on two lines of work. The first focuses on learning specialized representations for the visible portion of the occluded object, while the second explores contextual information to boost detection of occluded objects.

Modeling partial objects. Humans can recognize objects even from on small discriminative parts (e.g. the wheel of a car). This observation has inspired work on finding different ways of preventing noisy image evidence (e.g. occluders) from impacting detection confidence in a negative way. Wang *et al.* (2009) have proposed one of the first methods addressing detection of partial people. Relying on a HOG-LBP descriptor (Ahonen *et al.*, 2006) to discover occluded object regions, this work presents a combination of global object and local part models in order to boost the score of the human hypothesis based on the visible portion of the person. Similarly, Wojek *et al.* (2011); Meger *et al.* (2011) explore using half-person detectors in the context of 3D scene understanding, showing that explicit occlusion handling results in improved 3D scene understanding. Furthermore, Desai and Ramanan (2012), realizing that articulated people tend to self-occlude their parts, introduce part representations with encoded self-occlusion reasoning in the context of human pose estimation. Going one step further, Wohlhart *et al.* (2012) introduce an ISM inspired joint segmentation and people detection method that explicitly addresses partial object detection. Tackling crowded scenes, the method consists of a conditional random field jointly reasoning about partial and whole object hypotheses in a given image.

Going beyond detecting partial humans, Vedaldi and Zisserman (2009) introduce a structured output max-margin framework with explicit handling of truncated objects. By dedicating a visibility term to different patches in the object template and

[3] J. Malik, invited talk, CVPR'12

using an explicit truncation mask (specifying the truncated cells via the visibility grid) the method showed large gains in bicycle and horse detection on the Pascal VOC dataset. Girshick *et al.* (2011) introduce an object detection grammar with explicit part-level occlusion reasoning. Combined with a discriminative structured prediction learning framework, although primarily designed for people detection, the method of Girshick *et al.* (2011) shows state-of-the-art results on the Pascal VOC dataset. In the context of autonomous driving, Xiang *et al.* (2015b) focus on the car class and introduced 3D occlusion patterns specifying, on the level of 3D voxels, which portions of the object are occluded, truncated and visible. The method leverages characteristic occlusion patterns in a driving scenario like cars parked on the side of the road, and learns voxel level occlusion statistics for different patterns.

Contextual occlusion models. The notion that multiple visual entities that occlude each other can possibly be beneficial for recognition has mostly arisen from the perspective of context-modeling. Small objects have been demonstrated to be easier to detect in the presence of larger ones that can be detected more reliably, Karlinsky *et al.* (2010) detect musical instruments leveraging the presence of people in images. Sports tools have been shown to enable easier human pose estimation and vice versa (Yao and Fei-Fei, 2010), groups of people hint on the presence of individuals (Eichner and Ferrari, 2010; Yang *et al.*, 2012), and frequent arrangements of objects have been shown to support identification of individual objects (Li *et al.*, 2012).

Only recently, Tang *et al.* (2012) and Tang *et al.* (2014) leveraged the joint appearance of multiple people for robust people detection and tracking by training a double-person detector (Felzenszwalb *et al.*, 2010) on pairs of people rather than single humans. When a two person bounding box is detected, a regression (Felzenszwalb *et al.*, 2010) step follows, providing two person hypotheses, one for the occluder and one for the occluded person. Tang *et al.* (2012) employed a sophisticated non-maxima suppression scheme, combining single object with double object hypotheses, resulting in excellent people detection performance on person tracking benchmarks. In a follow up, Tang *et al.* (2013) consider more general occlusion patterns mined from tracking data. By integrating the tracker in the person detector training, this work argues that the decision which occlusion patterns should be included in the detector should be done in tracking-aware fashion. They demonstrate that the tighter integration of the tracker into the detection learning leads to better tracking performance on several tracking benchmarks. Building on this idea, Arteta *et al.* (2013) introduce not just two, but multiple instance detectors for the tasks of cell detection in fluorescence microscopy images and standard pedestrian detection. Combining discriminative learning and dynamic programming inference on a tree structured region graph, Arteta *et al.* (2013) show that not only pedestrians, but also small objects like cells, which tend to form groups in microscopy images, can be reliably localized by exploiting contextual information. Similarly, Wu *et al.* (2015b) focus on a multi-car detection scenario and learn an And-Or graph capable of learning and detecting multiple cars. In addition, the model can represent part-level occlusion information resulting in a rich hierarchical model that can jointly detect

clusters of cars, disambiguate the individual cars in the clusters and finally reason about part visibility. Representing context has also been extensively explored for multi-object tracking (Xiang *et al.*, 2015a; Choi *et al.*, 2013b; Leal-Taixe *et al.*, 2014) and collective activity recognition (Choi and Savarese, 2012; Choi *et al.*, 2014; Choi and Savarese, 2014).

Going further away from occlusion reasoning, contextual information has shown to be beneficial for different tasks. In the realm of deformable part models, Mottaghi *et al.* (2014) exploit local and global context to boost object detection performance on Pascal VOC. Zhu *et al.* (2015) boost object detections by exploiting image regions in the vicinity of a detection. A convnet representation is used to score the appearance of the candidate detection, but also the surrounding context, resulting in state-of-the-art detection performance. Sun *et al.* (2014) propose a framework for scene understanding that models both things (objects) and stuff (sky, grass) using a common representation. The joint representation allows for geometric and semantic constraints between things and stuff categories formulated in a graphical model. Relying on an efficient MAP inference method, this work illustrates that contextual representation leads to competitive object segmentation results on the Pascal segmentation dataset.

An entirely different avenue has been taken in the context of robotics applications, where prior distributions over expected occlusions can be analytically derived for heavily constrained indoor scenes (Hsiao and Hebert, 2012).

2.1.4 Convnet representations

Part-based models have been predominantly used to represent objects, combining hand engineered image representations like HOG (Dalal and Triggs, 2005), SIFT (Lowe, 2004), ShapeContext Belongie *et al.* (2000) with powerful statistical learning techniques, e.g. SVM (Cortes and Vapnik, 1995) or boosting (Freund and Schapire, 1997). Recently, end-to-end deep feature learning techniques (LeCun *et al.*, 2015) relying on large amounts of available training data have profoundly changed the computer vision world. Convolutional neural networks (LeCun *et al.*, 1998), a combination of filter convolutions learned with the back-propagation (LeCun, 1988) algorithm, have been at the very frontier of these advances. Applying convolutional, pooling, fully connected and rectified linear layers in a predefined consecutive order, the AlexNet (Krizhevsky *et al.*, 2012) convnet architecture has shown outstanding object recognition results on the ImageNet benchmark, showing 15% better results than the previous state-of-the-art models. Taking the AlexNet architecture as their key feature, the R-CNN (Girshick *et al.*, 2014) and the OverFeat (Sermanet *et al.*, 2014) methods have transferred the success to the object class detection task.

Although convnets provide powerful representations that can be easily reused in a wide range of applications, it is not well understood what makes these representations so successful and also what these representations actually capture. The usual word of wisdom when it comes to convents is that more data and bigger models are the key to making them work. However, there are many remaining questions as

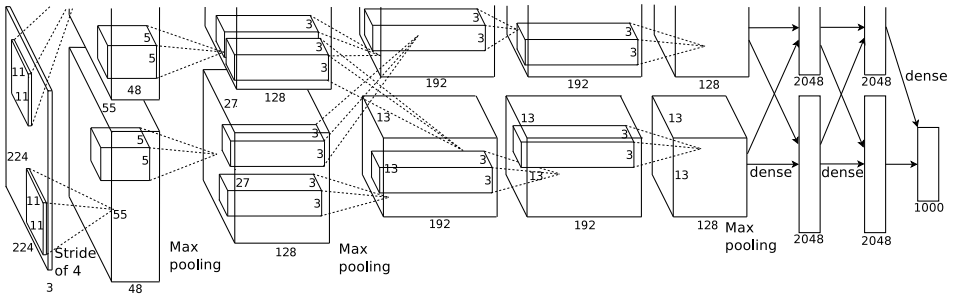


Figure 2.5: Convolutional neural networks. Figure from (Krizhevsky *et al.*, 2012).

e.g. how should the models grow, what kind of additional data helps most, how the object categories are represented internally and many more. Therefore, in this section we focus on previous work towards understanding convnets better. In particular, we are going to focus first on previous work on visualizing specific patterns emerging inside the network and second on work attempting to understand the convnet layer representations.

Understanding and visualizing convnet neurons. As neurons represent the building block of convnets, researchers have tried to isolate and understand the role of individual neurons. Driven by the intuition that grandmother neurons exist (those are neurons that fire whenever a concrete and specific concept is perceived, e.g. grandmother), many methods have attempted to visualize and understand individual neuron cells. Zeiler and Fergus (2014) introduce a visualization technique that reveals the input stimuli that excite individual neurons at any layer in the convnet architecture. Using a deconvolutional neural net (Zeiler *et al.*, 2011) to project individual neuron activations back to the image space, the method compares the different layers of an AlexNet architecture and shows the hierarchical nature of the features. While layer 2 features capture corner and edge-like image signals, layer 3 features tend to represent individual textures, mesh patterns and text in images. As one goes higher in the layers, higher conceptual things appear like animals and faces in layer 4. By visualizing the individual neuron responses at different layers, Zeiler and Fergus (2014) realize that the lower layers in AlexNET do not capture well the mid-range frequencies in the data, due to the structure of the filters. To that end, they propose smaller layer 1 and 2 filters with a smaller stride, resulting in performance gains over AlexNet. Furthermore, the work also illustrates the inability of convnets to handle partially visible objects. Following the same line of thought, Simonyan *et al.* (2014) established a mathematical connection between the deconvolutional networks and the neuron gradient computed w.r.t. an image. Using that, Simonyan *et al.* (2014) provide the first category level visualization technique, by finding the input image that gives the highest category specific score, for a fixed convnet architecture. Furthermore, when coupled with a given input image, the same technique is used for class specific saliency prediction, which in combination with a segmentation algorithm, is further used for weakly supervised object localization. While Simonyan *et al.* (2014) use L2-regularization, Yosinski *et al.*

(2015) use a natural image prior and also looked at not just the class neurons at the top, but also neurons inside the convnet. This again results in neuron visualizations suggesting that higher levels of abstractions are being built with depth and while lower layer neurons are responsible for localized image primitives like edges, and higher layer neurons respond to more semantically meaningful concepts. Following the direction of understanding individual neurons, Agrawal *et al.* (2014) study the existence of grandmother cells in specific layers of the AlexNet architecture, by computing the object localization performance using a subset of the convolutional filters. Their analysis illustrates that there exist a small number of grandmother-like features, but most of the feature code is distributed and in order to effectively discriminate between classes several neurons must jointly provide strong scores. On the other hand, Zhou *et al.* (2015) focus on the scene classification scenario and discovered that a large portion of the higher layer neurons do actually respond to specific object categories. However, again there is no single neuron dedicated to a specific category, but there is a set of neurons reacting to a single object category. Moreover, Zhou *et al.* (2015) illustrate that for scene classification, there is a minimal semantic input signal the convnet expects in order to be confident about an object category. For example, for a bedroom prediction, there has to be a bed and a window in the scene.

In the pursuit of understanding and visualizing convnets, Szegedy *et al.* (2014b) have taken a slightly different path. Instead of focusing on the output classifiers or individual inputs, they attempt to find adversary examples: minimal deviations of a given input signal that cause the convnet architecture to misclassify a given example. Their findings are interesting: by adding minimal noise to images, resulting in visual signal that is indistinguishable to humans, convnet networks provide random classification results, raising questions whether these representations comply to the one of the main assumptions in machine learning - the ideal problem representation should be smooth and the data intuitively lies on smooth manifolds. Nguyen *et al.* (2014) show a very related finding: it's easy to create images that are unrecognizable to humans, however convnets are strongly convinced about the categorization of such (to the average human) noisy images. This line of work on adversary examples raises interesting questions about the generalization ability of convnet architectures and the relation between human vision and convnet-based visual representations.

Understanding and visualizing convnet layers. Although individual neurons are important, convnet layers give the full representational power of convnet architectures. Therefore, understanding and visualizing these representations is a new and active research area. Mahendran and Vedaldi (2015) attempt to invert convnet representations from a specific layer by optimizing for the most likely image that would have generated the current response vector. Relying on natural image priors (total variation norm), the work illustrates that plausible image reconstructions can be obtained even from the highest layers of the convnet architecture. Exploring the inverted representations, the work illustrates that higher layer representations tend to discard low-level image statistics and also irrelevant transformations like translation and vertical rotation as these are irrelevant for high-level tasks. Observing

these effects Mahendran and Vedaldi (2015) illustrate that higher layers tend to be more invariant than lower layers. Dosovitskiy and Brox (2015) also attempt to invert convnet representations by having natural image prior. However, instead of solving an optimization problem, they train an up-convolutional neural net that reconstructs an image from an input activation vector. Their results lead to the same conclusions when it comes to invariances. The network becomes more invariant to transformations (especially translation and vertical rotation) with depth. However, Dosovitskiy and Brox (2015) illustrate that color information is preserved in the network layers. In fact, it turns out that there is a lot of information contained in the small class probabilities that are not among the top predictions of the network, which is in line with the "dark knowledge" idea of Hinton *et al.* (2015). Wei *et al.* (2015) also attempt to invert convnet representations, focusing on understanding inter-class variation. By relying on patch based natural image prior, this work attempts to organize natural image collections by discovering intrinsic, semantically meaningful variations. By analyzing the fully-connected layers in the AlexNet architecture, Wei *et al.* (2015) illustrate that the patch based prior results in more realistic and natural image reconstructions. In addition, this work shows that intra-class style information is organized in terms of location and content, represented in a hierarchical manner. While previous work attempts to invert a given convnet representation, Lenc and Vedaldi (2015) focus on learning various mappings and transformations with convnet layers. By learning such transformations, Lenc and Vedaldi (2015) can measure the equivariance and the equivalence of a given representation. The work illustrates that convnet representations change in an easily predictable manner with the input (under equivariant transformations). Contributing a transformation learning framework, this work illustrates that such equivariant mappings correspond to simple linear transformations of the convnet representations. Building on equivariances, Lenc and Vedaldi (2015) can also quantify the actual invariance to a given transformation, and quantitatively illustrate that invariances to flips, rotations and image rescaling grow with depth in the AlexNet architecture, which resonates with the work of Goodfellow *et al.* (2009), which also quantifies invariances in convnet architectures under certain transformations.

While the previous lines of work focus on inverting and quantifying invariances, a different line of work attempts to use synthetic image renderings to quantify properties of convnet representations. Using synthetic data has the advantage of a close-world where variations can be explicitly controlled. To that end, Aubry and Russell (2015) analyze convnet representations w.r.t. different appearance factors. Specifically, they take a convnet trained on ImageNet data and apply it to synthetically generated cars and chairs, quantifying the importance of different appearance factors in the representation variation. This allows for comparison across appearance factors. For example, Aubry and Russell (2015) show that in the fc7 layer of AlexNet, there is more variation in the features due to style, rather than viewpoint, confirming the viewpoint invariance of the model. Following a similar line of work, Peng *et al.* (2014) generate synthetic data for the Pascal VOC categories and show that even when using non-textured renderings convnets can

obtain reasonable performance. In addition, the work illustrates even when removing certain viewpoints from the training data, the overall detection performance is not significantly reduced, suggesting that the convnet representations are viewpoint invariant.

2.1.5 Relation to this thesis

In this section, we present the relation of this thesis and the presented related work, for each section separately, as far as general object class representations are concerned. The relation to 3D object and fine-grained representations is presented in Sections 2.2 and 2.3, respectively.

Part-based object representations. Most of the work in this thesis revolves around part-based object representations inspired by the deformable parts model (DPM) of Felzenszwalb *et al.* (2010). In contrast to the original DPM, which is tuned for object localization, in this work we introduce DPM versions tuned for richer object hypotheses. The DPMs presented in this work, first of all, require model structure which is specialized to the task at hand, both on a global object level, but also on a more local part level. While Felzenszwalb *et al.* (2010) introduce latent parts and components which correspond to different modes of the aspect ratio distribution, in our work we specialize the components to carry additional information like viewpoints or fine-grained labels. In addition, while the latent parts in the original DPM can be treated as additional features (Santosh K. Divvala, 2012), in our work we leverage object parts in several ways. We either constrain the parts to specific volumes of the 3D object geometry, resulting in 3D part representations, or rely on the parts' appearance, geometry and interplay to discriminate fine-grained categories. As we are interested in simultaneous reasoning about several tasks, representation learning has to reflect that as well. To that end, in contrast to the DPM of Felzenszwalb *et al.* (2010) which performs category model learning in a one-vs-all fashion, in this thesis we focus on multi-task joint learning of part-based models. Finally, as we exploit additional data sources like CAD models, the multi-task learning explored in this thesis is adaptive and reflects the fact that real world images should contribute towards learning realistic appearance, while the CAD models are a proxy towards learning about the 3D object geometry. We explore the relation to work on richer object representations in greater detail in Sections 2.2 and 2.3.

Occlusion representations. Concerning occlusion representations, in Chapter 9 we present a DPM inspired, occlusion-aware object class detector. In contrast to previous work focusing on capturing the visible part of the object only (Girshick *et al.*, 2011; Wang *et al.*, 2009; Wojek *et al.*, 2011; Meger *et al.*, 2011) and treating the occluded portion of the object as noise, in Chapter 9 we approach occlusion as first class citizen, by explicitly modeling the appearance of the occluder as well. While our occlusion aware representation is inspired by (Tang *et al.*, 2012) it differs in several ways. First, (Tang *et al.*, 2012) leverage the joint appearance of multiple people for robust people detection and tracking by training a rigid double-person

detector on pairs of people rather than single humans. While they considered only a rigid double person detector for side-ways walking people, we systematically evaluate and contrast different ways of modeling occluded-occlusion relations, and propose more expressive, hierarchical tree-like representation, as well a simple star-like occlusion representation. Second, while Tang *et al.* (2012) generate sideways walking double person examples, in our work we establish an automatic procedure for discovering relevant object-object occlusion cases in real world datasets (Geiger *et al.*, 2012), followed by a clustering procedure to discover characteristic object-object occlusion interactions. And third, in contrast to Tang *et al.* (2012) which focuses only on walking people, we consider a driving scenario with unconstrained cars and pedestrians. The idea has been recently revisited by Xiang *et al.* (2015b) in the context of learning 3D representations for car detection. In addition to our occlusion-aware representation, this work introduced voxelized 3D object representation that can be learned for different occlusion pattern. In a second stage, this work introduces a powerful graphical model that jointly reasons about the individual object hypotheses in a single image.

Convnet representations. The work presented in Chapter 10 is a step towards understanding deep convnet representations. Previous work has been focused either on visualizing and understanding individual neurons (Zeiler and Fergus, 2014; Simonyan *et al.*, 2014), attempting to establish relations with the human brain (Agrawal *et al.*, 2014), has focused on inverting convnet representations (Mahendran and Vedaldi, 2015; Dosovitskiy and Brox, 2015), or quantified the invariances in different layers (Lenc and Vedaldi, 2015; Goodfellow *et al.*, 2009). In contrast to previous work, we focus on analyzing what is holding back convnets to achieve better performance for object class detection. In particular, in the light of the usual word of wisdom that bigger models and more data always help, we attempt to understand what convnets have learned, and also what they actually can learn, by dissecting their performance across various appearance factors like viewpoints, shapes and sizes. In addition to recent work exploring synthetic data for convnet analysis (Aubry and Russell, 2015; Peng *et al.*, 2014) we introduce several novelties. First, we synthetically generate data for many object categories, in contrast to Aubry and Russell (2015) which focuses on cars and chairs only. Second, we investigate varying degrees of realism in the synthetically generated data, going from unrealistic wireframe renderings to renderings based on texture transfer. And third, in contrast to prior work, we sample the rendered data from the training data distribution of the real examples, which allows us to reason whether generating outliers like heavily truncated and extremely small objects is better than generating examples from the modes of the distribution.

2.2 3D OBJECT REPRESENTATIONS

While localization-oriented detectors are the de-facto state-of-the-art for object recognition, they provide very limited output, consisting of a bounding box (BB) and a class label, entirely ignoring the 3D nature of objects. On the other hand, models

with higher expressiveness have the potential to boost higher-level tasks like 3D scene understanding and autonomous driving by bridging the gap between the standard object detection output and the ideal input of these tasks. In contrast to the previous section where we explored object representations in general, in this section we focus on related work specializing on 3D object representations. Much of this work has been inspired by the 3D object representations from the early days of computer vision (see section 2.1). We gradually present the work in this area, first focusing on multi-view object representations, then proceeding with full 3D object representations that can reason not just about the viewpoint but also about the 3D shape of the object and finally providing a retrospective on the 3D scene understanding work which leverages 3D object representations.

2.2.1 Multi-view object detection

The interest in multi-view object representations has been inspired by the creation of several multi-view detection benchmarks: 3D object classes (Savarese and Fei-Fei, 2007), EPFL multi-view cars (Ozuysal *et al.*, 2009), Pascal3D+ (Xiang *et al.*, 2014a), KITTI (Geiger *et al.*, 2012). In this section we explore previous work on discrete and continuous viewpoint representations.

Discrete viewpoint representations. Since angular accurate viewpoint annotations are tedious and difficult to obtain, standard multi-view benchmarks typically provide categorical viewpoint annotations (Savarese and Fei-Fei, 2007; Lopez-Sastre *et al.*, 2010; Everingham *et al.*, 2010). This has inspired work on discrete multi-view object representations that usually model object classes as a collection of distinct views, forming a bank of viewpoint-dependent detectors. The specific form of these detectors is typically inspired by existing approaches from the literature that have proven to perform well for the single view case. One of the initial works in this area, Thomas *et al.* (2006), combine the implicit shape model of Leibe *et al.* (2004) and the multi-view framework of Ferrari *et al.* (2004) resulting in a successful multi-view object detector. In addition, the work connects parts across different views by means of feature tracking. Su *et al.* (2009) present a generative model for viewpoint estimation, triangulating the viewing sphere into discrete object viewpoints. By exploiting 3D geometric constraints the model establishes part correspondences across different views of the same object. Stark *et al.* (2010) on the other hand explore using non-photorealistic renderings of CAD models for multi-view model learning. Relying on a fully connected constellation model, connecting all parts in a given viewpoint, the viewpoint specific detectors are learned independently. Stark *et al.* (2010) show outstanding viewpoint estimation performance compared to the previous lines of work, on the relatively simple 3D Object Classes (Savarese and Fei-Fei, 2007) dataset. Instead of learning a bank of viewpoint detectors independently, another line of work considers joint training of a bank of viewpoint components. Gu and Ren (2010); Lopez-Sastre *et al.* (2011); Xiang *et al.* (2014a); Geiger *et al.* (2011) explicitly encode the viewpoint variable in the deformable parts model Felzenszwalb *et al.* (2010), by dedicating a component for each viewpoint bin. The joint multi-view training readily

outperforms the independent viewpoint learning methods (Gu and Ren, 2010) and makes a clear step towards challenging real world datasets like Pascal3D+ (Xiang *et al.*, 2014a) and KITTI (Geiger *et al.*, 2011). Payet and Todorovic (2011) use viewpoint specific shape templates that are subsequently matched to object contours in images. The highest scoring shape template determines the final object and viewpoint hypothesis. While most previous work has been focused on part-based representations, recently multi-view representations have also been popular within the deep learning community. Ghodrati *et al.* (2014) explore different feature representations for multi-view object detection, and illustrate that convnet features from an AlexNet model pre-trained on ImageNet can result in excellent joint object localization and viewpoint estimation. Following the same line of work, Tulsiani and Malik (2015) fine-tune an ImageNet pre-trained VGG architecture (Simonyan and Zisserman, 2015) for the same task, building an architecture where the output layer lives in the cross-product of classes and discrete viewpoint bins.

Discrete object viewpoint representations have been explored also in the context of high-level tasks. To that end, Geiger *et al.* (2011) rely on a 16-way viewpoint car detector that provides evidence for 3D scene understanding in videos. Similarly, Bao *et al.* (2012) explore viewpoint specific detectors for the task of object co-detection.

Continuous viewpoint representations. While multi-view approaches achieve remarkable results in predicting a discrete set of object poses, they have several limitations. First, they usually treat the discrete views independently (Lopez-Sastre *et al.*, 2011; Stark *et al.*, 2010; Gu and Ren, 2010). Second, they typically require evaluating a large number of view-based detectors, resulting in considerable runtime complexity (e.g., 32 shape templates (Payet and Todorovic, 2011), 36 constellation models (Stark *et al.*, 2010)). And third, the training data is typically split across the viewpoint detectors, resulting in only a few training examples per bin when high viewpoint resolution is required. Consequently, these methods do not scale towards fine-grained viewpoint estimation.

Therefore, researchers have explored building more scalable and continuous viewpoint representations. One of the prominent ideas in this direction is the multi-view object representation of Gu and Ren (2010) which in a first iteration estimates the coarse categorical viewpoint of the object, but in a second iteration refines the viewpoint hypothesis, a local gradient search around the coarsely estimated viewpoint. Modeling the continuous viewpoint as a first order Taylor expansion, the model learning is performed in a standard max-margin framework. The method showed state-of-the-art viewpoint estimation performance on the 3D object classes dataset (Savarese and Fei-Fei, 2007) at the time of publication.

Another line of research has been exploring the alignment of either coarse 3D shape models or detailed CAD models to objects in images. By jointly estimating the 3D shape and the camera parameters, these methods result in continuous and angular accurate viewpoint estimates. Therefore, in the next section we discuss 3D object models as object representations.

2.2.2 3D Object Models

As objects are inherently three-dimensional, researchers have explored 3D object representations even when relying only on a single image as input. Deemed very natural, 3D representations were the predominant paradigm back in the early days of computer vision. However, the difficulties arising from the ambiguities when matching 3D models to 2D evidence have diverted research towards more simplistic but robust 2D representations. Recently, researchers have revisited 3D representations either in the form of parametric, part-based representations or by considering collections of 3D CAD models. In this section, we provide an overview of state-of-the-art 3D representations, as well as methods used in the early days of computer vision.

3D implicit shape models. While Thomas *et al.* (2006) provide the fist multi-view ISM approach connecting codebook entries across neighboring views, Arie-Nachimson and Basri (2009) introduce one of the first 3D ISM versions. Instead of casting votes for object centers in image space, each matched codebook entry votes for the potential object center in 3D space. The 2D to 3D correspondences are driven by projective transformations of the 3D feature points to the image plane. The 3D representation is learned by applying a structure from motion technique to viewpoint registered training images. First, an instance specific 3D model is learned and then this model is enhanced with additional real world images of the object class of interest. At test time, the method defines a probability distribution parameterized with the 6 pose parameters (3 for rotation and translation), which is efficiently solved with a RANSAC procedure. This model showed modest car viewpoint estimation and detection performance on the 3D object classes (Savarese and Fei-Fei, 2007) and Pascal VOC 2007 (Everingham *et al.*, 2010) datasets. Glasner *et al.* (2011) follow a similar path as Arie-Nachimson and Basri (2009) with several modifications. First, the training data is extended to 22 car instances, and for each instance a 3D point cloud is obtained via SfM. The obtained point clouds are then manually aligned. In addition, their method contributes a faster and highly engineered 3D voting scheme which is in a next stage combined with SVM-HOG viewpoint specific classifiers which refine the viewpoint estimates. The work shows that discriminatively trained components are the key towards better performance. In addition, the method showed outstanding car viewpoint estimation performance on the Pascal VOC 2007 and 3D object classes datasets.

Sun *et al.* (2013) introduce a depth-encoded hough voting method for object detection and pose estimation in single images. The proposed method relies on depth data to guide the part selection process at training time. The resulting patches are not only supposed to be discriminative, but also have to result in physically plausible configurations. At test time, using the depth-encoded hough voting scheme, the model can prune the votes which result in physically inaccurate object constellations. The paper showed extensive experimental evaluation on several RGB and RGB-D datasets: a newly proposed 3D table-top dataset, ETHZ shapes, 3D object classes and Pascal 2007 dataset, showing promising and sometimes superior pose estimation

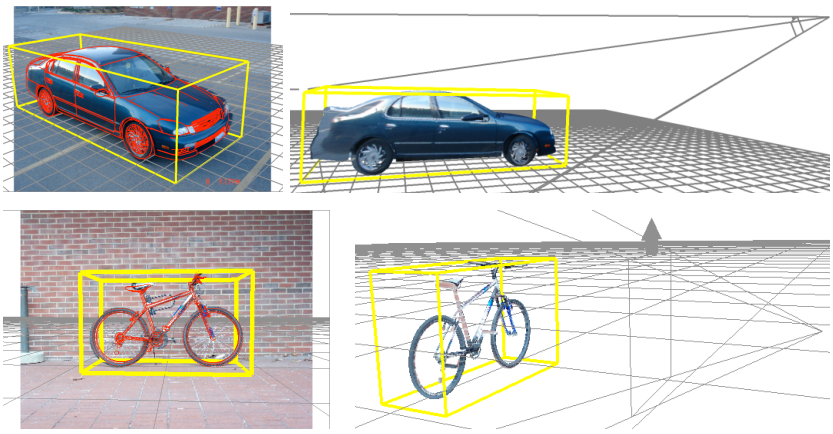


Figure 2.6: Detailed 3D wireframe object representation. Figure from Zia *et al.* (2013a)

.

and object detection performance.

3D constellation models. Apart from the 3D ISM models, other part-based models have also been explored in the past as a basis for 3D object representations. Zia *et al.* (2011) present a fully-connected 3D wireframe part-based model. At training time, CAD data is used to learn a PCA-based 3D shape model, while the 2D appearance terms, a combination of ShapeContext image descriptors and boosting classifiers, are learned using wireframe renderings of CAD models (Stark *et al.*, 2010). At recognition time, a two stage procedure is used. First, an object detector (Felzenszwalb *et al.*, 2010) provides a small set of candidate object detections, and then in a second step the 3D wireframe model is matched to the part score maps using a MCMC (Andrieu *et al.*, 2003) sampling procedure. The method obtained good performance on car localization and viewpoint estimation on the 3D object classes dataset. In addition, the 3D representation is also used in an ultra-wide baseline matching experiment, where given two wide baseline images of the same object the goal is to compute the fundamental matrix. The fundamental matrix is estimated by independently localizing object parts in the two images and then solving a linear system of equations. This work reliably matched parts across views, even when they are far apart. In Zia *et al.* (2013a), which is an extension of the previous work, in addition to cars, the evaluation is performed on bicycles from the 3D object classes dataset. Instead of boosting, random forests are used to learn the 2D part appearance terms. In addition to the 3D object classes, the work showed excellent viewpoint estimation performance on the relatively simple EPFL multi-view cars dataset (Ozuysal *et al.*, 2009). In another extension, Zia *et al.* (2013b) introduce a 3D wireframe model with occlusion reasoning. The work focuses on cars and uses a predefined set of characteristic car occlusion masks. In addition to showing improved quality on occluded objects, the work contributes a dataset of typical car occlusions.

Liebelt and Schmid (2010) take a slightly different path towards learning 3D object representations. While the previous line of work relies on CAD data to learn

about 3D geometry and 2D appearance terms, Liebelt and Schmid (2010) split the appearance and geometry learning apart. While realistic 2D image data is used to learn the part appearance terms, a combination of dense image features (Tola *et al.*, 2010) and spatial pyramids (Lazebnik *et al.*, 2006), the CAD data is used to learn strictly about the 3D object geometry, which is represented as part-specific mixtures of Gaussian distributions. In addition, while individual object parts in Zia *et al.* (2011) are manually annotated, in this work the parts are unsupervised and are obtained by splitting images into regular grids. The alignment of 2D and 3D data is on the level of viewpoints and is obtained by rendering the 3D CAD data in the same canonical views as the 2D data. Detection is again a two stage procedure where for a given object region, first the 2D part detectors are applied in a voting scheme resulting in part and rough viewpoint estimates. In a second stage, the 3D object pose and the camera parameters are estimated jointly in an expectation maximization (EM) (Dempster *et al.*, 1977) manner. The model achieved moderate viewpoint estimation and object localization performance on the 3D object classes dataset.

Yoruk and Vidal (2013) contribute a 3D object model composed of a set of 3D edge primitives, represented with their 3D position and 3D orientation. The edge-based 3D model can be viewed as a 3D generalization of HOG features. At training time, the model is learned from 2D image blueprints visualizing an object instance from a small but complementary set of viewpoints. A simple structure from motion method drives the 3D model reconstruction for each object instance, which in turn is used together with the reconstructions of all instances in a k-means clustering algorithm providing the final 3D set of primitives. At test time, the work relies on modified HOG features that focus on the distinct edges in an image, suppressing the noisy edge signals. These features are subsequently matched to a 3D object model in a branch-and-bound heuristics that aligns the 3D model in terms of its 3D orientation and 3D position via an orthographic camera model. Although interesting and fast, the proposed approach achieves only moderate performance levels in terms of object localization and viewpoint estimation on the 3D object classes and EPFL multi-view car datasets.

Deformable Part Models. While work discussed so far focused on densely connected part-based representations, next we will review models with simpler object topology, e.g. DPM (Felzenszwalb *et al.*, 2010) inspired star-like topology. Fidler *et al.* (2012) present a deformable 3D cuboid extension of DPM, enabling localization of objects with tight 3D bounding cubes. This works revisits the idea of modeling object aspects (Dickinson *et al.*, 1992). The aspects represent characteristic faces of the object of interest (e.g. frontal and side view of a car). In addition, this work strictly represents the appearance of a view in a fronto-parallel coordinate system, in an attempt to neutralize the effect of viewpoint variation on appearance. In addition to the rigid appearance of the whole object, each object view represents the appearance of latent parts, which are allowed to move w.r.t. the center of the object. The different views are connected via a stitching point on a 3D cuboid. At training time, the deformable 3D cuboid faces are trained in a joint max-margin

framework, similar in spirit to the one in Felzenszwalb *et al.* (2010). At test time, each test image is first rectified in the canonical fronto-parallel viewpoint of each face and the corresponding view detector is applied to that image. The different face detectors are combined with a max pooling operation. This work shows excellent recognition results in localizing objects on KITTI (Geiger *et al.*, 2012) but also in localizing objects in 3D on the indoor scene dataset of Hedau *et al.* (2010). Pedersoli and Tuytelaars (2014) recognize that at test time the image rectification step in Fidler *et al.* (2012) is a computational bottleneck and therefore propose a faster version where the images are not actually rectified, but the features of a particular viewpoint are computed as a linear combination of the features in the fronto-parallel views. In addition, the 3D cuboid in this work is not deformable, as object parts are not explicitly represented. The work reports speedups and comparable performance to standard multi-view detectors on the EPFL multi-view cars dataset (Ozuysal *et al.*, 2009) and the annotated faces in the wild (AFW, Zhu and Ramanan (2012)) dataset.

The work of M.Hejrati and D.Ramanan (2012) also explores using DPMs in the context of 3D object recognition. Their approach consists of two distinct stages. In the first stage, a DPM-inspired 2D object model for cars is learned, but in addition to the global mixtures typical for the classical DPM, the work of M.Hejrati and D.Ramanan (2012) also introduces local part mixtures that can reason about the visibility of parts but also capture varying part appearance. At test time, the DPM is applied densely over the image, resulting in part and whole object hypotheses. In the second stage, a 3D shape model of a car is aligned to each 2D object hypothesis. The 3D shape model is a collection of 3D basis shapes and is obtained using a non-rigid structure from motion technique, learned from annotated 2D correspondences. The connection of 2D observations and the 3D model are obtained via a weak-perspective projection model. At test time the 3D alignment step solves for a linear combination of the 3D basis shapes explaining the geometric part configuration observed in the image plane. This work provides coarse shape estimation as the 3D estimated shape is piecewise planar. The work reports moderate viewpoint estimation results on the EPFL multi-view cars dataset.

Xiang and Savarese (2012) present an aspect layout 3D object model, representing objects again with a piecewise planar 3D object representation, perfectly suited for convex and box-like objects like beds and cars. Xiang and Savarese (2012) set a different path when it comes to modeling object parts. Instead of having axis-aligned latent 3D parts, this work focuses on so called *aspect parts*: a planar portion of the objects whose entire 3D surface is readily visible in a given object viewpoint. This definition is similar to the aspect defined in Dickinson *et al.* (1992) and the aspect part definition in Koenderink and van Doorn (1979). The aspect layout model (ALM) represents aspect parts in 3D relying on a set of CAD models of the object class of interest. The aspect parts are manually annotated on the 3D models and in the 2D images. At training time, the 2D part unary and pairwise terms are learned in a max-margin framework and the 3D models are used to learn about the 3D object geometry and topology, by regularizing the part pairwise terms. At test time, for each viewpoint the search for optimal part placements for each part is done

separately. The highest scoring viewpoint provides the final hypothesis. This work reported very good object localization and viewpoint estimation results on several datasets: 3D object classes, EPFL multi-view cars and Pascal VOC cars 2006. In a next step, Xiang *et al.* (2014b) illustrate that 3D aspect parts can be successfully employed for monocular object tracking under significant viewpoint variations. In order to handle the large appearance variation due to viewpoint change, this work focuses on tracking individual 3D aspect parts. To enable robustness to occlusions, instance-level online part appearance learning is used. Via a car tracking experimental evaluation, the work demonstrates promising tracking and viewpoint estimation performance on a new car tracking benchmark, as well as on KITTI. Following the same line of research, Xiang and Savarese (2013) introduce the concept of 3D aspectlets - a planar collection of finer and more detailed atomic 3D aspect parts. The finer part representation employed in this work, allows for finer occlusion and visibility reasoning. The 3D aspectlets are used in this work for simultaneous 3D detection of multiple objects in an image, and the experimental evaluation on newly proposed indoor and outdoor datasets shows excellent 2D and 3D detection performance the the 3D aspectlets-based model.

Xiang *et al.* (2015b) introduce the notion of 3D voxel exemplars for object detection. A 3D voxel exemplar is a tuple of object in an image (e.g. a bounding box) and a 3D voxelized CAD model, where for every voxel a discrete variable describes whether the voxel is visible, occluded, self-occluded or truncated. The voxel representation allows for very fine-grained distinction (pixel level in image plane and voxel level in 3D) about the occupancy states of the object. At training time, a characteristic clusters of 3D voxel exemplars are found and 2D voxel detectors are trained using aggregated channel features (ACF, Dollár *et al.* (2014)) for each 3D voxel pattern. At test time, the trained detectors are applied first, and in a second step the meta-data (segmentation mask, 3D shape, truncation and occlusion labels) associated to each 3D voxel pattern detector are transferred to the object hypothesis. At the third and last step, given a camera model, a conditional random field reasons in 3D about the location, orientation and occlusion for each object hypothesis. The joint reasoning resulted in excellent car detection and viewpoint estimation performance on the KITTI (Geiger *et al.*, 2012) dataset.

Joint object reconstruction and recognition. Work discussed so far used 3D information in different forms (CAD data, multiple-views of the same instance) for the purpose of 3D recognition. Vicente *et al.* (2014) take a different path and aim at providing category-level 3D reconstructions without using any 3D information a priori. The reconstruction procedure uses class labels, bounding boxes, segmentation masks and semantic keypoint annotations to reconstruct the object category. Relying on a structure from motion techniques, Vicente *et al.* (2014) assume that all instances of the same class can be treated as if they are the same instance in the SfM algorithm. The actual reconstruction is a two-stage procedure. First, using a rigid SfM over the ground truth keypoints a rough viewpoint is recovered for every instance for the category of interest. The procedure also results in a rough 3D shape estimate, which in this case is represented by the set of keypoints lifted to 3D. After computing the

average category 3D shape, in a next step the shape of each object in the database is estimated, borrowing from the shape information available in the other instances of the same class. A visual hull optimization scheme is used for the second step, optimizing over the segmentation and the available 2D keypoints. The work reported coarse but satisfying reconstructions of Pascal VOC as well as on a synthetically generated dataset. However, the major drawback is that the method uses ground truth information only. To overcome this problem, Kar *et al.* (2015) introduced a 3D object reconstruction method that combines object reconstruction with shape from shading. At training time, a coarse 3D category shape is learned and the individual viewpoints of all object instances are estimated using 2D keypoint annotations in a non-rigid SfM (Bregler *et al.*, 2000). Then, using 2D segmentation masks and sub-category annotations, a more detailed shape model is learned for every sub-category (e.g. car types). At test time, the reconstruction is obtained in a forward process, where after recognizing the category, segmenting the object and estimating the viewpoint, shape from shading (Barron and Malik, 2015) is used to recover high frequencies in the depth map of the object of interest. While interesting in spirit, the work also provides a quantitative evaluation of the quality of the estimated 3D shapes, and shows small gains w.r.t. Vicente *et al.* (2014).

Bao *et al.* (2013) introduce semantic shape priors for multi-view object reconstruction. Standard structure from motion methods result in sparse and coarse 3D category reconstructions. This work results in much more detailed reconstructions relying on category-specific shape priors. Relying on 3D scans of objects, (Bao *et al.*, 2013) learn an accurate category-level 3D shape along with so-called 3D anchor points which capture 3D geometry, but also 2D object appearance. At test time, after the initial sparse multi-view reconstruction, the anchor points are used to align the 3D category shape to the current object, but also to drive the 3D shape refinement to the current instance. This work shows encouraging results on a newly introduced multi-view and 3D scanner data dataset, containing 3 categories: car, fruit and keyboard.

3D instance alignment. Due to the availability of large amounts of free 3D CAD models, recently researchers have been exploring alignment of CAD models to objects in images. The accurate alignment of CAD models results not only in full 3D viewpoint estimation, but also in very accurate and detailed shape estimation. Aubry *et al.* (2014) present a robust and fast method to align 3D CAD models of chairs. At the core of the alignment step lies an exemplar oriented, part-based 3D object representation that consists of a large amount (800K) of local discriminative parts. At training time, all of the CAD models are rendered in a predetermined and densely sampled set of views. For each view, a discriminative subset of parts is chosen which are next used to learn discriminative exemplar LDA (Hariharan *et al.*, 2012) classifiers on HOG features. These classifiers are fast and easy to obtain, however their scores are not well calibrated and therefore in a next step the scores are calibrated using a validation set of negative image patches. During recognition, the large set of discriminative elements is densely applied over a given image and a simple star-like spatial model reasons about the plausible configurations of parts from a single

rendering. The found part detections are accumulated and the highest scoring configuration provides the aligned CAD model rendering, which in turn provides the meta data (viewpoint and shape) information. The method showed moderate performance on Pascal VOC 2007 chairs class, focusing only on the non-occluded and non-truncated chairs.

While Aubry *et al.* (2014) focus on chairs in realistic images, Huang *et al.* (2015) explore detailed 3D reconstructions of simplistic images of furniture, illustrating e.g. a chair on a fairly clean background. The work is based on a joint analysis over images and CAD models. While Aubry *et al.* (2014) strictly align a CAD model from a pre-defined set, in this work, if a 3D CAD example does not match the shape of the 2D observation, it can be altered accordingly. Therefore, the proposed method can also address noisy 3D shapes, frequently encountered in 3D repositories. The optimization procedure proceeds in several stages. First, all CAD models are densely rendered in 360 viewpoints. Then, a coarse 3D viewpoint is estimated for each 2D image by solving a structured prediction problem, optimizing over the real and the rendered images. Using HOG features as image representation, the goal of this step is to find close rendered images in appearance, which are then used to transfer the viewpoint and the shape information. In a next step, dense pixel-level correspondences are established across patches of real world and rendered images using HOG and SIFTflow (Liu *et al.*, 2011) features in a spectral clustering algorithm. Finally, using the dense pixel correspondences both image domains are jointly segmented and subsequently a new 3D model is synthesized for every 2D image, via an optimization problem reusing the matched parts of existing components and enforcing smoothing constraints. The method shows very good reconstruction and pose estimation performance on simplistic images only.

Kholgade *et al.* (2014) devise an interactive photo-editing tool which allows object manipulation in a 3D scene using a single image. The method has several stages. First, the user selects an object in the image and a 3D model and manually provides a rough, but fairly accurate alignment of the 3D model and the 2D object. Then, a semi-automatic procedure provides masks for the foreground object (including the shade), the ground plane and the background pixels. The method then estimates the illumination and adjusts the 3D geometry to the observed object in the scene. The user can then manipulate the object in the scene, while the method automatically renders the object realistically in the scene, estimating the correct illumination and filling up the missing object pixels (due to previous self-occlusion in the original pose).

While Kholgade *et al.* (2014) rely on an interactive approach, Rematas *et al.* (2014) automatically align 3D CAD models to images, which in turn are used to generate images of objects in novel views with consistent appearance, shape and illumination. The method starts from a coarse viewpoint alignment of the object in the image and the 3D CAD model. The object in the image is then deformed to match the shape of the rendered 3D object. In a next step, the object in the image is rendered in a novel view, and each pixel in the new image is reconstructed via an optimization problem assigning the appearance of a pixel from the original view based on the matching

probability of the two pixels. Rematas *et al.* (2014) illustrate that synthesizing novel views can be used to enhance existing training datasets, showing improved detection performance on Pascal VOC 2007, especially on the rare viewpoint cases. Further, this work shows that the approach can be used for re-synthesis, in order to denoise, inpaint, and upsample images as well as generate stereo-pairs from single images.

Choy *et al.* (2015) revisit the idea of exemplar-based detectors for the task of fast and accurate alignment of CAD models to images. While standard object detectors have an offline learning stage, the goal of this work is to generate renderings and train exemplar LDA detectors on the fly, allowing for scalable 3D object detection applications. The key component of this work is the NZ-WHO (non-zero whitened histogram of orientations), a well engineered and faster version of the WHO (Hariharan *et al.*, 2012) image descriptor, employing conjugate gradient for faster solving of linear systems, and Fast Fourier Transform (FFT)-based filter convolution. At test time, the method uses Markov Chain Monte Carlo (MCMC) sampling to drive the iterative fitting of a 3D model to images. Combining the proposed approach with a state-of-the-art object localization method, R-CNN (Girshick *et al.*, 2014), this work achieves excellent object localization and viewpoints estimation performance for cars and bicycles on the Pascal3D+ (Xiang *et al.*, 2014a) and 3D object classes datasets.

2.2.3 3D Scene Understanding

As 3D scene understanding methods implicitly assume richer object representations, we give a brief overview of the most prominent outdoor scene understanding and indoor scene understanding methods.

Outdoor 3D scene understanding. Wojek *et al.* (2010, 2013) introduce one of the first monocular 3D scene understanding approaches. Relying on a video as input, the method predicts 3D object positions, object tracks, ground plane and horizon line. Using object detection (Wojek *et al.*, 2009) and semantic segmentation labels (Wojek and Schiele, 2008) as input across frames, the method first estimates object tracklets in the videos using a probabilistic graphical model. Then, in a second step, the object tracklets, in combination with 3D priors, constraining the height and the position of the specific instances of a given category, the whole scene is tracked using a hidden Markov model (HMM), relying on MCMC inference. The work illustrated that scene level reasoning leads to improved object detection performance on several datasets.

Geiger *et al.* (2011, 2014) present a monocular 3D understanding method, predicting the scene topology, geometry and the scene activities (object tracks) using a short video sequence. While similar to Wojek *et al.* (2010) in its input, the method also represents the scene category (4 different topology related categories seen from bird-eye perspective), as well as a flexible scene geometry model, allowing for adaptive road topologies. This is driven by a vanishing point representation with two points, one corresponding to the road the moving platform is on, and another one, corresponding to the orthogonal road, if present in the scene, according to the scene topology. Inference is also done by MCMC sampling. The method showed that scene-level reasoning leads to better object viewpoint estimation, as well as vanishing

point detection.

Wang *et al.* (2015) explore using geographic priors for simultaneous semantic segmentation, depth estimation and 3D object detection from a single image. While the previous line of work reasons about the scene topology, in this work the geographical location is used to generate a detailed, static 3D world of buildings and roads given the current object location. Given the 3D world, the algorithm has to reason only about the dynamic elements in the scene. To that end, a pixel-level conditional random field is build, representing the semantic label and depth of the pixel. In addition, the CRF captures 3D objects in the scene, represented with CAD models of the corresponding class of interest. At test time, inference is done in a block coordinate descent fashion optimizing for one task at a time. The method is employed on the KITTI dataset showing state-of-the-art depth and semantic labeling performance.

Zia *et al.* (2014b) present a 3D scene understanding method that instead of using a simplistic 2D bounding box or 3D cube as object representation, relies on a high resolution object representation (Zia *et al.*, 2013a). As a follow up of the single 3D object representation framework (Zia *et al.*, 2011), this work uses only a single view as input and reasons not just about the 3D position and orientation of a single object but jointly about a multitude of objects including reasoning about their interactions as well as the ground plane. The 3D scene reasoning allows for part-level occlusion reasoning, which ultimately led to good localization and viewpoint estimation performance on KITTI.

Indoor scene understanding. Inspired by indoor scene datasets like (Hedau *et al.*, 2010), NYU Depth (Nathan Silberman and Fergus, 2012) and SUN3D (Xiao *et al.*, 2013), a lot of research has focused on understanding indoor scenes. Here we give a rather broad overview of the most prominent directions in indoor scene understanding.

Wang *et al.* (2010a) explore the problem of estimating the 3D layout of a cluttered room. Assuming a Manhattan world, the task boils down to inferring three vanishing points, rendering the room layout as a 3D cube. While the 3D cube is the desired output, indoor scenes usually contain static furniture, decorations and dynamic elements like people, which in this work are treated as latent clutter that has to be inferred. To that end, an over-segmentation method is used and each segment is assigned a binary clutter variable. At training time, the method uses a structured output max-margin formulation. At test time, a stochastic hill climbing approach is used iterating between the box layout inference and the clutter estimation. The method showed moderate improvements on the indoor clutter dataset (Hedau *et al.*, 2009). Furlan *et al.* (2013) focus on using video sequences rather than single images, for 3D layout estimation. While most of indoor scene understanding literature assumes Manhattan worlds, this work eliminates that assumption, which ultimately allows for much more flexible layout estimates.The method proposes a 3D layout estimation pipeline. In the first step, the video sequence is used to obtain camera location and sparse 3D reconstruction of the scene. In a second step, the 3D points are used to generate candidate layout elements (floors, ceilings and walls). In

the final step, the layout elements are combined to obtain final layout estimation, relying on a probabilistic layout estimation framework combined with particle filter inference. The work showed very good performance on a newly proposed indoor scenes dataset.

In contrast to Wang *et al.* (2010a) who reason only about the scene layout, Schwing *et al.* (2013) propose a joint 3D room layout and 3D object estimation approach. Constraining the search to a single object, the method combines top-down object location and viewpoint evidence from (Fidler *et al.*, 2012; Felzenszwalb *et al.*, 2010) with bottom-up image features, geometric context (Hedau *et al.*, 2009) and orientation maps (Lee *et al.*, 2009). The method presents an energy formulation combining a data fitness function defined on the room layout and the dominant object in the room, and a regularization terms biasing the solution towards simple explanations and favoring geometrically plausible solutions. Model learning is again performed in a structure output max-margin formulation and at test time an efficient and novel branch-and-bound algorithm is used. The experimental evaluation on the bedrooms dataset (Hedau *et al.*, 2010) illustrates improved object localization due to scene-level reasoning as well as moderate improvements on layout estimation.

Choi *et al.* (2013a) propose a hierarchical graphical model jointly addressing object detection, layout estimation and scene classification for 3D indoor scene understanding. At the core of their method are the so called 3D geometric phrases (3DGP) - 3D constellations of commonly occurring objects (e.g. a table and chairs), describing 3D relations among objects. The 3DGPs allow the model to detect partially visible objects like chairs and side tables, relying on contextual information. Contributing with a new dataset combining the three tasks, (Choi *et al.*, 2013a) demonstrate that the 3DGPs and the joint reasoning about the tasks leads to excellent performance in terms of scene classification, object detection and layout estimation.

As for indoor scenes commodity depth sensors can be used, Lin *et al.* (2013) present a method for indoor scene understanding with RGB-D data. The proposed method jointly estimates semantic labels, scene geometry and places 3D cuboids around objects. Following a hierarchical representation this work contributes a CRF with the scene type defining the object-object and object-scene interactions all the way down to 3D cuboid representation of the objects. As input, this work relies on a 3D object proposal method. For that purpose a constrained parametric min-cuts (CPMC)-based (Carreira and Sminchisescu, 2012) object proposal method is used, modified to handle RGB-D data. The experimental evaluation on NYU Depth (Nathan Silberman and Fergus, 2012) shows convincing 3D object recognition performance.

In the realm of indoor scene understanding with RGB-D data, Kim *et al.* (2013) introduced a voxel-based 3D scene representation, where essentially for each voxel the method infers if it belongs to the current scene (voxelized scene geometry) and the semantic label associated to it. The work proposes a voxel-CRF defined over a 3D voxel grid, capturing the semantic labels of each voxel, while at the same time reconstructing the scene in 3D, a virtue of using depth data. Each voxel has an occupancy, semantic class and visibility variable associated to it. The voxel CRF

captures per voxel observations from the image and depth channels, and imposes smooth pairwise constraints, as well as enforces 3D surface (Borrmann *et al.*, 2011) and object-level (Felzenszwalb *et al.*, 2010) constraints. Graph-cuts are used at test time for inference. The experimental evaluation on NYU Depth suggests that the method achieves moderate semantic segmentation improvements.

2.2.4 Relation to this thesis

This section highlights the relation of this thesis to different 3D object representations. In particular, we focus on each individual group of methods: multi-view representations, 3D object models and 3D scene understanding.

Multi-view object representations. In chapters 3 and 5 we present deformable parts model based, multi-view object representations. While previous DPM multi-view methods (Gu and Ren, 2010; Lopez-Sastre *et al.*, 2011; Geiger *et al.*, 2011; Xiang *et al.*, 2014a) learn models with one-vs-all classification loss, we introduce a structured output multi-task loss, jointly optimizing for object localization and viewpoint estimation, resulting in state-of-the-art joint object localization and viewpoint estimation performance on 3D object classes, EPFL multi-view car, KITTI and Pascal3D+ dataset. In addition, while state-of-the-art part-based multi-view representations learn parts independently across viewpoints, our multi-view detectors in chapters 3 and 4 can establish part correspondences across multiple viewpoints, as we rely on 3D information when learning the models. We illustrate that our multi-view detectors can establish reliable part-correspondences across two views of the same object, even when the views are far apart.

3D object representations. In chapters 4 and 5 we present a 3D deformable part model, which is related to the work of Fidler *et al.* (2012). However, our 3D DPM represents both the object and the parts in 3D space and not just their position but also the part pairwise terms. In addition, our 3D DPM model employs continuous appearance model, allowing for arbitrary fine viewpoint estimates. The model achieves excellent localization and viewpoint estimation performance not just on simple, but on challenging datasets like KITTI outperforming the standard DPM. This is in contrast to most 3D object models, which traditionally have been hard to match to 2D image evidence. In Chapter 6 we present a 3D object detection method aligning CAD models to objects in images. While previous work (Aubry *et al.*, 2014; Rematas *et al.*, 2014) typically focused on fully visible chairs and cars, in this work we "go in the wild" and apply our method to multiple object categories with difficult examples including small and occluded objects. Our method achieved state-of-the-art 3D object detection performance on Pascal3D+.

3D scene understanding. State-of-the-art outdoor scene understanding methods require stereo systems and video streams. In addition, they rely on a simplistic object representations either in the form of simple bounding boxes (Wojek *et al.*, 2010), multi-view representations (Geiger *et al.*, 2011) or 3D cubes (Schwing *et al.*, 2013). Our 3D object representations are more detailed, representing 3D part posi-

tions and deformations (see Chapter 5) as well as detailed 3D shape information (see Chapter 6). Furthermore, we do not require 3D part annotations for our 3D DPM part based models, in contrast to Zia *et al.* (2014b).

2.3 FINE-GRAINED REPRESENTATIONS

In this section we explore fine-grained object representations. The methods in this area assume that the object has already been localized and focus on disambiguating the subordinate affiliation. Progress in this field has been driven by several well established datasets: CUB-200 dataset of bird species (Welinder *et al.*, 2010b), Stanford cars dataset (Stark *et al.*, 2012), Oxford flowers (Nilsback and Zisserman, 2008) and aircrafts (Maji *et al.*, 2013) datasets. While this field is quite broad on its own, we focus on work that is strongly related to the work presented in this thesis, primarily in terms of part-based representations. According to Tversky and Hemenway (1984), the presence or absence of parts is related to the formation of basic-level object categories (a car has wheels, a chair does not), while specific properties of parts are indicative of subordinate categories (a sports car has a different trunk than a sedan). Therefore, we explore related work driven by this principle.

Leveraging humans in the loop. As localization and disambiguation of discriminative and local part information is a difficult and challenging task, using humans has been actively explored, either directly in the inference loop, or to get useful detailed annotations at scale. To that end, Branson *et al.* (2010) present a crowd-sourced question-answering framework to describe detailed part information which otherwise would be hard to detect and represent. Given an image of a bird the system asks questions about certain details (e.g. color or shape of certain parts like the belly or the beak) which the user has to answer. While the pool of candidate questions in each iteration is large, the goal is to choose the question which maximizes the information gain. With each iteration, the class probability distribution is adapted to the answer of the human, taking into account the answer confidence. SVM classifiers on global SIFT features are used to obtain the initial class distribution. The method has been applied on the CUB-200 birds dataset while the set of questions is extracted using a set of binary bird attributes. The work illustrates that computer vision features do help in their framework, however only for easy tasks (only a few questions are required to reach the correct answer).

Following the same idea of using human knowledge, Deng *et al.* (2013) attempt to learn discriminative bird features using crowdsourcing tools. Disambiguation of fine-grained categories often requires expert knowledge, which in turn limits large-scale fine-grained categorization benchmarks. To that end, this work proposes a gamified subordinate annotation approach - the bubble game. Each time a user plays the game a blurred image is shown, along with two images of two different species. The user has to decide to which of the two categories the bird in the blurred image belongs to. The user can click on certain parts of the blurred image, allowing him to see clearly the underlying image patch. The goal is to use as few clicks as

possible. In this way, the bubble game allows non-experts to be used at scale in order to obtain part-level discriminative annotations. While the set of all category pairs is large, the bubble game is played only for the classes which are highly similar, a measure obtained on a validation set. In a next step, the bubble annotations are used in a BubbleBank detector - a bank of exemplar bubble detectors. In the end, a linear one-vs-all SVM classifier is trained for each category, providing the final classification result. The work resulted in excellent fine-grained recognition performance on CUB-200.

3D representations. A few lines of work have focused on using 3D representations for fine-grained recognition, motivated by the need of pose-invariant representations addressing the large variability in poses and viewpoints, especially for the birds class. Farrell *et al.* (2011) present the birdlets approach - a method for pose normalized bird species categorization. Birdlets are Poselet-based (Bourdev and Malik, 2009) part detectors, which in addition to the 2D position, are also parameterized with 3D orientation and scale, constituting a 3D ellipsoid representation for parts. This work focuses on representing the head and the body of the bird category. While the 3D ellipsoids are manually annotated, at training time the method clusters the birdlets of all examples of a given species, resulting in very specialized volumetric birdlet representation. At test time, the birdlets are applied over the image resulting in ellipsoid-like detections in the image. Then, the ellipsoids are used for viewpoint and pose-invariant object representation, consisting of cropped patches from the aligned ellipsoids. SIFT features are extracted from the ellipsoid patches and a random forest (Breiman, 2001) classifier is trained on top for fine-grained categorization. The method is applied on the CUB-200 dataset, resulting in moderate fine-grained recognition performance.

Krause *et al.* (2013) explore using 3D information for fine-grained recognition of car types (Stark *et al.*, 2012). Their method provided the first 3D variants of two popular representations in fine-grained recognition and object categorization in general. Using 3D information from CAD models, this work provided a 3D version of the BubbleBank (Deng *et al.*, 2013) representation, as well as the Spatial Pyramid matching (SPM) (Lazebnik *et al.*, 2006) descriptor. The essential difference to their 2D counterparts is that the spatial pooling at test time in this work is performed on a 3D volume. Relying on CAD data, the method first estimates the 3D geometry of the car, relying on a set of CAD models. Matching is driven by a HOG-SVM classifier trained for each car type and viewpoint. After precise alignment, geometry driven appearance is calculated both for the 3D BubbleBank and the 3D SPM method. The method has been applied to the Stanford fine-grained cars dataset (Stark *et al.*, 2012), resulting in improvements over the 2D counterparts, with the 3D BubbleBank method being better than the 3D SPM.

Convnet representations. Zhang *et al.* (2014) explored using the R-CNN (Girshick *et al.*, 2014) detection framework for fine-grained recognition. The method first detects object parts in the image and then extracts convnet features for the parts which are fed into a fine-grained recognition method. Relying on manual part annotations, at training time whole-object and part classifiers are trained using the

R-CNN framework. To that end the AlexNet convnet architecture is tuned for the task of object and part detection. At test time first the object and part detectors are applied on a set of candidate object and part regions. For each object detection, the part locations are estimated using simplistic 2D geometric topologies. Then, the part-features are extracted and fed into a one-vs-all SVM trained on fine-grained recognition. The method achieved state-of-the-art performance on the CUB-200 dataset.

Krause *et al.* (2015) also rely on convnet representations, but acknowledge that part annotations are tedious and prevent fine-grained categorization at scale. To that end, they propose a fine-grained recognition method without any part annotation, based on co-segmentation. At training time, the work attempts to establish part-correspondences across images of the same fine-grained category by co-segmenting (based on GrabCut (Rother *et al.*, 2004)) two images with similar poses. Then, the dense pixel correspondences are propagated in a graph, where images with similar poses are only connected. After obtaining a diverse set of parts from the dense set of correspondences via k-means part-trajectory clustering, part-detectors are trained similar in nature to Zhang *et al.* (2014). In addition, this work explores applying co-segmentation of detected objects, by co-segmenting them with the most similar object in the training set. The method showed state-of-the-art performance on the CUB-200 birds classification dataset, as well as on the Stanford cars dataset.

Applications. While previous lines of work treat fine-grained categorization in isolation, Mottaghi *et al.* (2015) introduce a hierarchical coarse-to-fine model for joint 3D pose estimation and fine-grained categorization, recognizing that these tasks are highly correlated. To address that, this work proposes a 3-layered hierarchical model, where at the highest layer is the object category, and as one goes towards the leaves more fine-grained categories are introduced. While the viewpoint variable on the base-category level is discrete, on the second and third layers it attains a continuous angular representation of the azimuth, the elevation and the distance of the object to the camera. The proposed model also represents object shape, using a CAD model per category in the lowest (third) layer. The model is formulated with an energy function relying on HOG and convnet features as appearance representation and imposing categorical, pose and shape constraints across the hierarchy. At training time, the model is learned in a max-margin structured SVM framework. At test time, after discretizing the continuous variables, for a given candidate object detection (provided by the R-CNN) the inference can be performed via exhaustive search due to the small state-space. The work achieves excellent object viewpoint estimation and sub-category recognition performance on the car, airplane and boat categories of the Pascal3D+ dataset.

2.3.1 Relation to this thesis

In this section, we give an overview of the relation of this thesis to the presented fine-grained categorization methods.

Part-based representations. The fine-grained recognition method presented in Chapter 7 is inspired by the fact that part appearance and geometry encode subordinate association. However, unlike most of the fine-grained representations, in this work we do not rely on part-level annotations and furthermore we model object viewpoints jointly with fine-grained categories. In addition, while most methods use one-vs-all independent learning of category specific classifiers, we introduce joint multi-class learning of fine-grained categories. While all fine-grained benchmarks contain a balanced training and test set for all categories, we acknowledge that the real-world distributions are unbiased and skewed, and therefore in Chapter 8 we address learning multi-view subordinate models from sparse viewpoint data.

Applications. Similar to Mottaghi *et al.* (2015) in Chapter 8 we illustrate that fine-grained information can be useful for generic object class detection, providing higher overall precision than base-level object detectors. Furthermore, in Chapter 7 we explore the usability of fine-grained information in the context of 3D scene understanding and illustrate that subordinate information enables more detailed object distance to camera estimation.

TEACHING 3D GEOMETRY TO DEFORMABLE PART MODELS 3

Contents

3D geometry plays a vital role in building representative object descriptions. Therefore, in the following chapters (chapters 3, 4, 5, 6), we introduce object representations with gradually increasing level of 3D geometric detail. In this chapter in particular, we focus on robust multi-view object representations that represent object parts in 3D. Using the deformable parts model (Felzenszwalb *et al.*, 2010) as a starting point, we gradually *teach* it about 3D object geometry, resulting in representations that not only localize objects in images, but can reliably estimate viewpoints and establish part correspondences across different object views. In an extensive experimental evaluation on real world datasets, we verify the excellent viewpoint and part-correspondence estimation capabilities of the proposed representations.

3.1 INTRODUCTION

Object class recognition has reached remarkable performance for a wide variety of object classes, based on the combination of robust local image features with statistical learning techniques (Fergus *et al.*, 2003; Leibe *et al.*, 2004; Felzenszwalb *et al.*, 2010). Success is typically measured in terms of 2D bounding box (BB) overlap between hypothesized and ground truth objects (Everingham *et al.*, 2010) favoring algorithms implicitly or explicitly optimizing this criterion (Felzenszwalb *et al.*, 2010).

At the same time, interpretation of 3D visual scenes in their entirety is receiving increased attention. Reminiscent of the earlier days of computer vision (Marr and Nishihara, 1978; Brooks, 1981; Pentland, 1986; Lowe, 1987), rich, 3D geometric

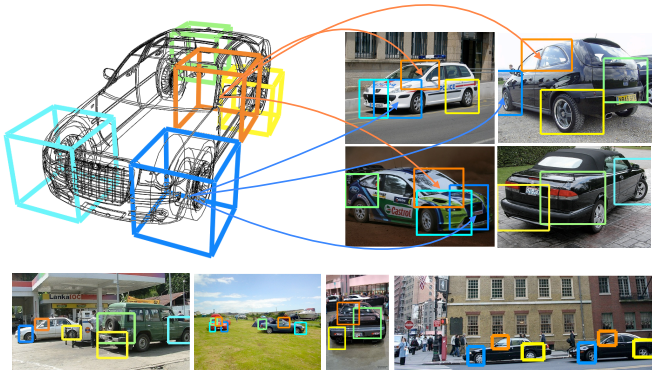


Figure 3.1: Example detections of our DPM-3D-Constraints. Note the correspondence of parts found across different viewpoints (color coded), achieved by a 3D parameterization of latent part positions (left). Only five parts (out of 12 parts) are shown for better readability.

representations in connection with strong geometric constraints are increasingly considered a key to success (Hoiem *et al.*, 2008; Ess *et al.*, 2009; Gupta *et al.*, 2010; Wang *et al.*, 2010a; Wojek *et al.*, 2010; Bao and Savarese, 2011; Gupta *et al.*, 2011). Strikingly, there is an apparent gap between these rich 3D geometric representations and what current state-of-the-art object class detectors deliver. As a result, current scene understanding approaches are often limited to either qualitative (Gupta *et al.*, 2010) or coarse-grained quantitative geometric representations, where reasoning is typically limited to the level of entire objects (Hoiem *et al.*, 2008; Gupta *et al.*, 2010; Wang *et al.*, 2010a; Wojek *et al.*, 2010).

The starting-point and main contribution of this chapter is therefore to leave the beaten path towards 2D BB prediction, and to explicitly design an object class detector with outputs amenable to 3D geometric reasoning. By basing our implementation on one of the most renowned 2D BB-based object class detector to date, the deformable parts model (DPM (Felzenszwalb *et al.*, 2010)), we ensure that the added expressiveness of our model comes at minimal loss with respect to its robust image matching to real images. To that end, we propose to successively add geometric information to our object class representation, at three different levels.

First, we rephrase the DPM as a genuine structured output prediction task, comprising estimates of both 2D object BB *and* viewpoint. This enables us to explicitly control the trade-off between accurate 2D BB localization and viewpoint estimation. Second, we enrich part and whole-object appearance models by training images rendered from CAD data. While not being as representative as real images in terms of feature statistics, these images literally come with perfect 3D annotations e.g. for position and viewpoint, which we can use to improve localization performance and viewpoint estimates.

And third, we extend the notion of discriminatively trained, deformable parts to 3D, by imposing 3D geometric constraints on the latent positions of object parts. This ensures consistency between parts across viewpoints (i.e., a part in one view

corresponds to the exact same physical portion of the object in another view, see Fig. 3.1), and is achieved by parameterizing parts in 3D object coordinates rather than in the image plane during training. This consistency constitutes the basis of both reasoning about the spatial position of object parts in 3D and establishing part-level matches across multiple views. In contrast to prior work based on 3D shape (Stark *et al.*, 2010; Zia *et al.*, 2011), our model learns 3D volumetric parts fully automatically, driven entirely by the loss function.

In an experimental study, we demonstrate two key properties of our models. First, we verify that the added expressive power w.r.t accurate object localization, viewpoint estimation and 3D object geometry does not hurt 2D detection performance too much, and even improves in some cases. In particular, we first show improved performance of our structured output prediction formulation over the original DPM for 18 of 20 classes of the challenging Pascal VOC 2007 data set (Everingham *et al.*, 2007). We then show that our viewpoint-enabled formulation further outperforms, to the best of our knowledge, all published results on 3D Object Classes (Savarese and Fei-Fei, 2007).

Second, we showcase the ability of our model to deliver geometrically more detailed hypotheses than just 2D BBs. Specifically, we show a performance improvement of up to 8% in viewpoint classification accuracy compared to related work on 9 classes of the 3D Object Classes data set. We then exploit the consistency between parts across viewpoints in an ultra-wide baseline matching task, where we successfully recover relative camera poses of up to 180 degrees spacing, again improving over previous work (Zia *et al.*, 2011).

3.2 STRUCTURED OUTPUT LEARNING FOR DPM

In the following, we briefly review the DPM model (Felzenszwalb *et al.*, 2010) and then move on to the extensions we propose in order to "teach it 3D geometry". For comparability we adopt the notation of (Felzenszwalb *et al.*, 2010) whenever appropriate.

3.2.1 DPM review

We are given training data $\{(I_i, y_i)\}_{1,\dots,N}$ where I denotes an image and $y = (y^l, y^b) \in \mathcal{Y}$ is a tuple of image annotations. The latter consists of y^b, the BB position of the object, e.g. specified through its upper, lower, left and right boundary, and $y^l \in \{-1, 1, \dots, C\}$ the class of the depicted object or -1 for background.

A DPM is a mixture of M conditional random fields (CRFs). Each component is a distribution over object hypotheses $z = (p_0, \dots, p_n)$, where the random variable $p_j = (\mathrm{u}_j, \mathrm{v}_j, l_j)$ denotes the (u, v)-position of an object part in the image plane and a level l_j of a feature pyramid image features are computed on. The root part p_0 corresponds to the BB of the object. For training examples we can identify this with y^b, whereas the parts $p_1, \dots, p_n$ are not observed and thus latent variables. We

collect the two latent variables of the model in the variable $h = \{c, p_1, \ldots, p_n\}$, where $c \in \{1, \ldots, M\}$ indexes the mixture component.

Each CRF component is star-shaped and consists of unary and pairwise potentials. The unary potentials model part appearance as HOG (Dalal and Triggs, 2005) template filters, denoted by $F_0, \ldots, F_n$. The pairwise potentials model displacement between root and part locations, using parameters (v_j, d_j), where v_j are anchor positions (fixed during training) and d_j a four-tuple defining a Gaussian displacement cost of the part p_j relative to the root location and anchor. For notational convenience we stack all parameters in a single model parameter vector for each component c, $\beta_c = (F_0, F_1, \ldots, F_n, d_1, \ldots, d_n, b)$, where b is a bias term. We denote with $\beta = (\beta_1, \ldots, \beta_M)$ the vector that contains all parameters of all mixture components. For consistent notation, the features are stacked $\Psi(I, y, h) = (\psi_1(I, y, h), \ldots, \psi_M(I, y, h))$, with $\psi_k(I, y, h) = [c = k]\psi(I, y, h)$, where $[\cdot]$ is Iverson bracket notation. The vector $\Psi(I, y, h)$ is zero except at the c'th position, so we realize $\langle \beta, \Psi(I, y, h) \rangle = \langle \beta_c, \psi(I, y, h) \rangle$. The un-normalized score of the DPM, that is the prediction function during test-time, solves $\operatorname{argmax}_{(y,h)} \langle \beta, \Psi(I, y, h) \rangle$.

3.2.2 Structured max-margin training (DPM-VOC)

The authors of (Felzenszwalb *et al.*, 2010) propose to learn the CRF model using the following regularized risk objective function (an instance of a latent-SVM), here written in a constrained form. Detectors for different classes are trained in a one-versus-rest way. Using the standard hinge loss, the optimization problem for class k reads

$$\begin{aligned} \min_{\beta, \xi \geq 0} \quad & \frac{1}{2}\|\beta\|^2 + C \sum_{i=1}^{N} \xi_i \\ \text{sb.t.} \quad & \forall i : y_i^l = k : \max_h \langle \beta, \Psi(I_i, y_i, h) \rangle \geq 1 - \xi_i \\ & \forall i : y_i^l \neq k : \max_h \langle \beta, \Psi(I_i, y_i, h) \rangle \leq -1 + \xi_i. \end{aligned} \tag{3.1}$$

While this has been shown to work well in practice, it is ignorant of the actual goal, 2D BB localization. In line with (Blaschko and Lampert, 2008) we hence adapt a structured SVM (SSVM) formulation using margin rescaling for the loss, targeted directly towards 2D BB prediction. For a part-based model, we arrive at the following, latent-SSVM, optimization problem

$$\begin{aligned} \min_{\beta, \xi \geq 0} \quad & \frac{1}{2}\|\beta\|^2 + C \sum_{i=1}^{N} \xi_i \\ \text{sb.t.} \quad & \forall i, I_i, \bar{y} \neq y_i : \max_{h_i} \langle \beta, \Psi(I_i, y_i, h_i) \rangle \\ & - \max_h \langle \beta, \Psi(I_i, \bar{y}, h) \rangle \geq \Delta(y_i, \bar{y}) - \xi_i \end{aligned} \tag{3.2}$$

$$\tag{3.3}$$

where $\Delta : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}_+$ denotes a loss function. Like in (Blaschko and Lampert, 2008) we define $\Psi(I, y, h) = 0$ whenever $y^l = -1$. This has the effect to include the

two constraint sets of problem (3.1) into this optimization program.

Based on the choice of Δ we distinguish between the following models. We use the term DPM-Hinge to refer to the DPM model as trained with objective (3.1) from (Felzenszwalb *et al.*, 2010) and DPM-VOC for a model trained with the loss function

$$\Delta_{\text{VOC}}(y, \bar{y}) = \begin{cases} 0, & \text{if } y^l = \bar{y}^l = -1 \\ 1 - [y^l = \bar{y}^l] \frac{A(y \cap \bar{y})}{A(y \cup \bar{y})}, & \text{otherwise} \end{cases} \tag{3.4}$$

first proposed in (Blaschko and Lampert, 2008). Here $A(y \cap \bar{y}), A(y \cup \bar{y})$ denote the area of intersection and union of y^b and $\bar{y}^b$.The loss is independent of the parts, as the BB is only related to the root.

Training We solve (3.2) using our own implementation of a gradient descent with delayed constraint generation. The latent variables render the optimization problem of the DPM a mixed integer program and we use the standard coordinate descent approach to solve it. With fixed β we find the maxima of the latent variables h_i for all training examples and also search for new violating constraints $\bar{y}, h$ in the training set. Then, for fixed latent variables and constraint set, we update β using stochastic gradient descent.

Note that the maximization step over h involves two latent variables, the mixture component c and part placements p. We search over c exhaustively by enumerating all possible values $1, \ldots, M$ and for each model solve for the best part placement using the efficient distance transform. Similar computations are needed for DPM-Hinge. Furthermore we use the same initialization for the anchor variables v and mixture components as proposed in (Felzenszwalb *et al.*, 2010) and the same hard negative mining scheme.

3.3 EXTENDING THE DPM TOWARDS 3D GEOMETRY

As motivated before, we aim to extend the outputs of our object class detector beyond just 2D BB. For that purpose, this section extends the DPM in two ways: a) including a viewpoint variable and b) parametrizing the entire object hypothesis in 3D. We will refer to these extensions as a) DPM-VOC+VP and b) DPM-3D-Constraints.

3.3.1 Introducing viewpoints (DPM-VOC+VP)

Our first extension adds a viewpoint variable to the detector output, which we seek to estimate at test time. Since several real image data sets (e.g., 3D Object Classes (Savarese and Fei-Fei, 2007)) as well as our synthetic data come with viewpoint annotations, we assume the viewpoint observed during training, at least for a subset of the available training images. We denote with $y^v \in \{1, \ldots, K\}$ the viewpoint of an object instance, discretized into K different bins, and extend the annotation accordingly to $y = (y^l, y^b, y^v)$.

We allocate a distinct mixture component for each viewpoint, setting $c = y^v$ for

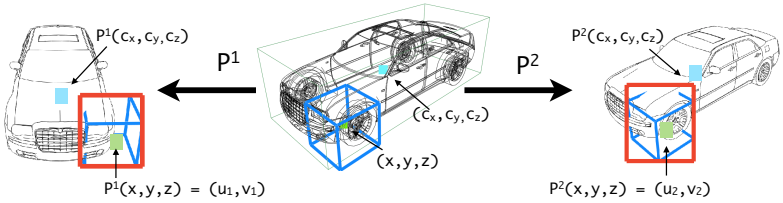


Figure 3.2: 3D part parametrization for an example 3D CAD model (center). Corresponding projected part positions in 2 different views, overlaid non-photorealistic renderings (Stark *et al.*, 2010) (left, right).

all training examples carrying a viewpoint annotation. During training, we then find the optimal part placements for the single component matching the annotation for the training examples (which speeds up training). For training examples where a viewpoint is not annotated we proceed with standard DPM training, maximizing over components as well. At test time we output the estimated mixture component as a viewpoint estimate.

Since the component-viewpoint association alone does not yet encourage the model to estimate the correct viewpoint (because Eq. (3.3) does not penalize constraints that yield the correct BB location but a wrong viewpoint estimate), we exploit that our objective function is defined for general loss functions. We add a $0/1$ loss term for the viewpoint variables, in the following convex combination

$$\Delta_{\text{VOC+VP}}(y, \bar{y}) = (1 - \alpha)\Delta_{\text{VOC}}(y, \bar{y}) + \alpha \left[y^v \neq \bar{y}^v \right]. \tag{3.5}$$

Note that any setting $\alpha \neq 0$ is likely to decrease 2D BB localization performance. Nevertheless we set $\alpha = 0.5$ in all experiments and show empirically that the decrease in detection performance is small, while we gain an additional viewpoint estimate. Note also, that the constraint set from Eq. (3.3) now include those cases where the BB location is estimated correctly but the estimated mixture component (in h) does not coincide with the true viewpoint.

A less powerful but straight-forward extension to DPM-Hinge is to use the viewpoint annotations as an initialization for the mixture components, which we refer to in our experiments as DPM-Hinge-VP.

3.3.2 Introducing 3D parts (DPM-3D-Constraints)

The second extension constitutes a fundamental change to the model, namely, a parametrization of latent part positions in 3D object space rather than in 2D image coordinates. It is based on the intuition that parts should really live in 3D object space rather than in the 2D image plane, and that a part is defined as a certain partial volume of a 3D object rather than as a 2D BB.

We achieve this by basing our training procedure on a set of 3D CAD models of the object class of interest that we use in addition to real training images. Being formed from triangular surface meshes, 3D CAD models provide 3D geometric

descriptions of object class instances, lending themselves to 3D volumetric part parametrizations. The link to recognizing objects in 2D images is established by projecting the 3D parts to a number of distinct viewpoints, "observed" by viewpoint dependent mixture components, in analogy to DPM-VOC+VP. Since all components observe the parts through a fixed, deterministic mapping (the projections), their appearances as well as their deformations are *linked* and kept consistent across viewpoints by design.

3D Parametrization. Given a 3D CAD model of the object class of interest, we parametrize a part as an axis-aligned, 3D bounding cube of a fixed size per object class, $p_j = (x_j, y_j, z_j)$, positioned relative to the object center (its root, see Fig. 3.2), in analogy to positioning parts relative to the root filter in 2D for DPM-Hinge. Further, we assume a fixed anchor position for each part p_j, from which p_j will typically move away during training, in the course of maximizing latent part positions h.

Model structure. The DPM-3D-Constraints consists of a number of viewpoint dependent mixture components, and is thus structurally equivalent to the DPM-VOC+VP. Each component observes the 3D space from a specific viewpoint c, defined by a projective mapping P^c. In full analogy to the DPM-VOC+VP, for each part p_j, each component observes i) part appearance as well as ii) part displacement. Here, both are uniquely determined by the projection $P^c(p_j)$. For i), we follow (Stark *et al.*, 2010) to generate a non-photorealistic, gradient-based rendering of the 3D CAD model, and extract a HOG filter for the part p_j directly from that rendering. For ii), we measure the displacement between the projected root and the projected part position. Part's displacement distribution is defined in the image plane and it is independent across components. As a short-hand notation, we include the projection into the feature function $\Psi(I_i, y_i, h, P^c)$.

Learning. Switching to the 3D parametrization requires to optimize latent part placements h over object instances (possibly observed from multiple viewpoints simultaneously) rather than over individual images. Formally, we introduce an object ID variable y^o to be included in the annotation y.

For a training instance y^o, we let $S(y^o) := \{i : y_i^o = y^o\}$ and compute

$$h^* = \operatorname*{argmax}_h \sum_{i \in S(y^o)} \left\langle \beta, \Psi(I_i, y_i, h, P^{y_i^v}) \right\rangle . \tag{3.6}$$

This problem can be solved analogously to its 2D counterpart DPM-VOC+VP: assuming a fixed placement of the object root (now in 3D), we search for the best placement h of each of the parts also in 3D. The score of the placement then depends simultaneously on all observing viewpoint-dependent components, since changing h potentially changes all projections. The computation of the maximum is still a linear operation in the number of possible 3D placements, and we use the same optimization algorithm as before: alternate between a) updating β and b) updating h and searching for violating constraints. Note that the DPM-3D-Constraints introduces additional constraints to the training examples and thereby lowers the number of free parameters of the model. We attribute performance differences to DPM-VOC+VP to this fact.

AP	aero	bird	bike	boat	bottle	bus	car	cat	cow	table	dog
DPM-Hinge	30.4	1.8	61.1	13.1	**30.4**	50.0	63.6	9.4	30.3	17.2	1.7
DPM-VOC	31.1	2.7	**61.3**	14.4	29.8	**51.0**	**65.7**	12.4	32.0	19.1	2.0
Vedaldi *et al.*	**37.6**	**15.3**	47.8	**15.3**	21.9	50.7	50.6	**30.0**	**33.0**	**22.5**	**21.5**

AP	horse	mbike	person	plant	sheep	sofa	train	tv	chair	AVG
DPM-Hinge	56.5	48.3	42.1	6.9	16.5	26.8	43.9	37.6	18.5	30.3
DPM-VOC	**58.6**	**48.8**	**42.6**	7.7	20.5	27.5	43.7	38.7	**18.7**	31.4
Vedaldi *et al.*	51.2	45.5	23.3	**12.4**	**23.9**	**28.5**	**45.3**	**48.5**	17.3	**32.1**

Table 3.1: 2D bounding box localization performance (in AP) on Pascal VOC 2007 (Everingham *et al.*, 2007), comparing DPM-Hinge, DPM-VOC, and (Vedaldi *et al.*, 2009). Note that (Vedaldi *et al.*, 2009) uses a kernel combination approach that makes use of multiple complementary image features.

Blending with real images. Training instances for which there is only a single 2D image available (e.g., Pascal VOC data) can of course be used during training. Since there are no other examples that constrain their 3D part placements, they are treated as before in (3.2). Using real and synthetic images for training is called *mixed* in the experiments.

Initialization. In contrast to prior work relying on hand-labeled semantic parts (Stark *et al.*, 2010; Zia *et al.*, 2011), we initialize parts in the exact data-driven fashion of the DPM, only in 3D: we choose greedily k non-overlapping parts with maximal combined appearance score (across views).

Self-occlusion reasoning. Training from CAD data allows to implement part-level self-occlusion reasoning effortlessly, using a depth buffer. In each view, we thus limit the number of parts to the l ones with largest visible area.

3.4 EXPERIMENTS

In this section, we carefully evaluate the performance of our approach, analyzing the impact of successively adding 3D geometric information as we proceed. We first evaluate the 2D BB localization of our structured loss formulation, trained using only Δ_{VOC} (DPM-VOC, Sect. 3.2.2). We then add viewpoint information by optimizing for $\Delta_{\text{VOC+VP}}$ (DPM-VOC+VP, Sect. 3.3.1), enabling simultaneous 2D BB localization and viewpoint estimation. Next, we add synthetic training images (Sect. 3.3.2), improving localization and viewpoint estimation accuracy. Finally, we switch to the 3D parameterization of latent part positions during training (3D^{2}PM, Sect. 3.3.2), and leverage the resulting consistency of parts across viewpoints in an ultra-wide baseline matching task. Where applicable, we compare to both DPM-Hinge and

results of related work.

3.4.1 Structured learning

We commence by comparing the performance of DPM-VOC to the original DPM (DPM-Hinge), using the implementation of (Felzenszwalb *et al.*, 2009). For this purpose, we evaluate on two diverse data sets. First, we report results for the detection task on all 20 classes of the challenging Pascal VOC 2007 data set (Everingham *et al.*, 2007). Second, we give results on 9 classes of the 3D Object Classes data set (Savarese and Fei-Fei, 2007), which has been proposed as a testbed for multi-view recognition, and is considered challenging because of its high variability in viewpoints (objects are imaged from 3 different distances, 3 elevations, and 8 azimuth angles). In all experiments, we use images from the respective data sets for training (sometimes in addition to our synthetic data), following the protocols established as part of the data sets (Everingham *et al.*, 2007; Savarese and Fei-Fei, 2007).

2D Bounding box localization. Tab. 3.1 gives results for 2D BB localization according to the Pascal criterion, reporting per-class average precision (AP). It compares our DPM-VOC (row 2) to the DPM-Hinge (Felzenszwalb *et al.*, 2009) (row 1) and to the more recent approach (Vedaldi *et al.*, 2009) (row 3), both of which are considered state-of-the-art on this data set. We first observe that DPM-VOC outperforms DPM-Hinge on 18 of 20 classes, and (Vedaldi *et al.*, 2009) on 8 classes. While the relative performance difference of 1.1% on average (31.4% AP vs. 30.3% AP) to DPM-Hinge is moderate in terms of numbers, it is consistent, and speaks in favor of our structured loss over the standard hinge-loss. In comparison to (Vedaldi *et al.*, 2009) (32.1% AP), DPM-VOC loses only 0.7% while the DPM-Hinge has 1.8% lower AP. We note that (Vedaldi *et al.*, 2009) exploits a variety of different features for performance, while the DPM models rely on HOG features, only.

Tab. 3.2 gives results for 9 3D Object Classes (Savarese and Fei-Fei, 2007), comparing DPM-Hinge (col. 1), DPM-VOC+VP (col. 3), and DPM-Hinge-VP (col. 2), where we initialize and fix each component of the DPM-Hinge with training data from just a single viewpoint, identical to DPM-VOC+VP. We observe a clear performance ordering, improving from DPM-Hinge over DPM-Hinge-VP to DPM-VOC+VP, which wins for 5 of 9 classes. While the average improvement is not as pronounced (ranging from 88.0% over 88.4% to 88.7% AP), it confirms the benefit of structured vs. hinge-loss.

Viewpoint estimation. Tab. 3.2 also gives results for viewpoint estimation, phrased as a classification problem, distinguishing among 8 distinct azimuth angle classes. For DPM-Hinge, we predict the most likely viewpoint by collecting votes from training example annotations for each component. For DPM-Hinge-VP and DPM-VOC+VP, we use the (latent) viewpoint prediction. In line with previous work (Savarese and Fei-Fei, 2007; Lopez-Sastre *et al.*, 2011), we report the mean precision in pose estimation (MPPE), equivalent to the average over the diagonal of the 8 (viewpoint) class confusion matrix. Clearly, the average MPPE of 87.1% of DPM-VOC+VP outperforms DPM-Hinge-VP (74.7%) and DPM-Hinge (55.8%). It

AP / MPPE	DPM-Hinge	DPM-Hinge-VP	DPM-VOC+VP
iron	94.7 / 56.0	93.3 / 86.3	**96.0** / **89.7**
shoe	95.2 / 59.7	**97.9** / 71.0	96.9 / **89.8**
stapler	82.8 / 61.4	**84.4** / 62.8	83.7 / **81.2**
mouse	**77.1** / 38.6	73.1 / 62.2	72.7 / **76.3**
cellphone	60.4 / 54.6	**62.9** / 65.4	62.4 / **83.0**
head	87.6 / 46.7	89.6 / 89.3	**89.9** / **89.6**
toaster	97.4 / 45.0	96.0 / 50.0	**97.8** / **79.7**
car	99.2 / 67.1	99.6 / 92.5	**99.8** / **97.5**
bicycle	97.9 / 73.1	98.6 / 93.0	**98.8** / **97.5**
AVG	88.0 / 55.8	88.4 / 74.7	**88.7** / **87.1**

Table 3.2: 2D bounding box localization (in AP) and viewpoint estimation (in MPPE (Lopez-Sastre *et al.*, 2011)) results on 9 3D Object classes (Savarese and Fei-Fei, 2007).

also outperforms published results of prior work (Lopez-Sastre *et al.*, 2011) (79.2%) and (Gu and Ren, 2010) (74.0%) by a large margin of 7.9%. Initializing with per-viewpoint data already helps (59.8% vs. 74.7%), but we achieve a further boost in performance by applying a structured rather than hinge-loss (from 74.7% to 87.14%). As a side result we find that the standard DPM benefits from initializing the components to different viewpoints. We verified that fixing the components does not degrade performance, this is a stable local minima. This makes evident that different viewpoints should be modeled with different mixture components. A nice side effect is that training is faster when fixing mixture components.

Summary. We conclude that structured learning results in a modest, but consistent performance improvement for 2D BB localization. It significantly improves viewpoint estimation over DPM-Hinge as well as prior work.

3.4.2 Extending DPMs towards 3D

3.4.2.1 *Synthetic training data*

In the following, we examine the impact of enriching the appearance models of parts and whole objects with synthetic training data. For that purpose, we follow (Stark *et al.*, 2010) to generate non-photorealistic, gradient-based renderings of 3D CAD models, and compute HOG features directly on those renderings. We use 41 cars and 43 bicycle models[4] as we have CAD data from these two classes only.

2D bounding box localization. We again consider Pascal VOC 2007 and 3D Object Classes, but restrict ourselves to the two object classes most often tested by prior work on multi-view recognition (Liebelt and Schmid, 2010; Stark *et al.*,

[4]www.doschdesign.com, www.sketchup.google.com/3dwarehouse/

Pascal 2007 (Everingham *et al.*, 2007)

	AP / MPPE	Glasner *et al.*	DPM-Hinge	DPM-VOC	DPM-3D-Constr.
cars	real	32.0	63.6	65.7	-
	synthetic	-	24.7	34.5	24.9
	mixed	-	65.6	**66.0**	63.1
bicycle	real	-	61.1	61.3	-
	synthetic	-	22.6	25.2	20.7
	mixed	-	60.7	**61.6**	56.8

3D Object Classes (Savarese and Fei-Fei, 2007)

Liebelt and Schmid	Zia *et al.*	Payet and Todorovic	Glasner *et al.*	DPM-Hinge	DPM-VOC+VP	DPM-3D-Constr.
-	-	- / 86.1	99.2 / 85.3	99.2 / 67.1	99.8 / 97.5	-
-	90.4 / 84.0	-	-	92.1 / 78.3	98.6 / 92.9	94.3 / 84.9
76.7 / 70	-	-	-	99.6 / 86.3	**99.9** / **97.9**	99.7 / 96.3
-	-	- / 80.8	-	97.9 / 73.1	**98.8** / 97.5	-
-	-	-	-	72.2 / 77.5	78.1 / 86.4	72.4 / 82.0
69.8 / 75.5	-	-	-	97.3 / 73.1	97.6 / **98.9**	95.0 / 96.4

Table 3.3: 2D bounding box localization (in AP) on Pascal VOC 2007 (Everingham *et al.*, 2007) (up) and 3D Object Classes (Savarese and Fei-Fei, 2007) (down). Viewpoint estimation (in MPPE (Lopez-Sastre *et al.*, 2011)) on 3D Object Classes (down). Top three rows: object class car, bottom three rows: object class bicycle.

2010; Payet and Todorovic, 2011; Glasner *et al.*, 2011; Zia *et al.*, 2011), namely, cars and bicycles. Tab. 3.3 (up) gives results for Pascal cars and bicycles, comparing DPM-Hinge (col. 2) and DPM-VOC (col. 3) with the recent results of (Glasner *et al.*, 2011) (col. 1) as a reference. We compare 3 different training sets, *real*, *synthetic*, and *mixed*. First, we observe that *synthetic* performs considerably worse than *real* in all cases, which is understandable due to their apparent differences in feature statistics. Second, we observe that DPM-VOC improves significantly (from 24.7% to 34.5% AP) over DPM-Hinge for *synthetic* on cars, highlighting the importance of structured training. Third, we see that *mixed* consistently outperforms *real* for DPM-VOC, obtaining state-of-the-art performance for both cars (66.0% AP) and bicycles (61.6% AP).

Tab. 3.3 (down) gives results for 3D Object Classes, again training from *real*, *synthetic*, and *mixed* data, sorting results of recent prior work into the appropriate rows. In line with our findings on Pascal, we observe superior performance of DPM-VOC+VP over DPM-Hinge, as well as prior work. Surprisingly, *synthetic* (98.6% AP) performs on cars almost on par with the best reported prior result (Glasner *et al.*, 2011) (99.2%). *Mixed* improves upon their result to 99.9% AP. On bicycles, the appearance differences between *synthetic* and *real* data are more pronounced, leading to a performance drop from 98.8% to 78.1% AP, which is still superior to

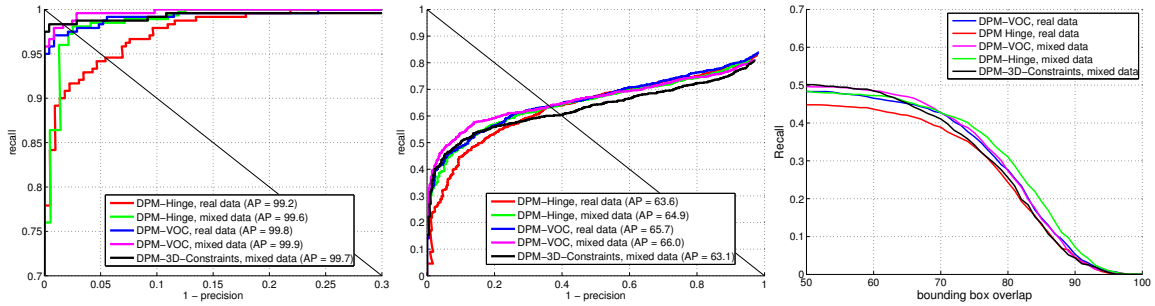


Figure 3.3: Detailed comparison of *real* and *mixed* training data. Left: Precision-recall on 3D Object Classes (Savarese and Fei-Fei, 2007) cars (zoomed). Middle: Precision-recall on Pascal VOC 2007 (Everingham *et al.*, 2007) cars. Right: Recall over bounding box overlap at 90% precision on Pascal 2007 cars.

DPM-Hinge *synthetic* (72.2% AP) and the runner-up prior result of (Liebelt and Schmid, 2010) (69.8% AP), which uses *mixed* training data.

In Fig. 3.3, we give a more detailed analysis of training DPM-Hinge and DPM-VOC from either *real* or *mixed* data for 3D Object Classes (Savarese and Fei-Fei, 2007) (left) and Pascal 2007 (Everingham *et al.*, 2007) (middle, right) cars. In the precision-recall plot in Fig. 3.3 (middle), DPM-VOC (blue, magenta) clearly outperforms DPM-Hinge (red, green) in the high-precision region of the plot (between 0.9 and 1.0) for both *real* and *mixed*, confirming the benefit of structured max-margin training. From the recall over BB overlap at 90% precision plot in Fig. 3.3 (right), we further conclude that for DPM-Hinge, *mixed* (green) largely improves localization over *real* (red). For DPM-VOC, *real* (blue) is already on par with *mixed* (magenta).

Viewpoint estimation. In Tab. 3.3 (down), we observe different behaviors of DPM-Hinge and DPM-VOC+VP for viewpoint estimation, when considering the relative performance of *real*, *synthetic*, and *mixed*. While for DPM-VOC+VP, *real* is superior to *synthetic* for both cars and bicycles (97.5% vs. 92.9% and 97.5% vs. 86.4%), the DPM-Hinge benefits largely from synthetic training data for viewpoint classification (improving from 67.1% to 78.3% and from 73.1% to 77.5%). In this case, the difference in feature statistics can apparently be outbalanced by the more accurate viewpoints provided by *synthetic*.

For both, DPM-Hinge and DPM-VOC+VP, *mixed* beats either of *real* and *synthetic*, and switching from DPM-Hinge to DPM-VOC+VP improves performance by 11.6% for cars and 25.8% for bicycles, beating runner-up prior results by 11.8% and 18.1%, respectively.

Summary. We conclude that adding synthetic training data in fact improves the performance of both 2D BB localization and viewpoint estimation. Using *mixed* data yields state-of-the-art results for cars and bicycles on both Pascal VOC 2007 and 3D Object classes.

3.4.2.2 *3D deformable parts*

We finally present results for the DPM-3D-Constraints, constraining latent part positions to be consistent across viewpoints. We first verify that this added geometric expressiveness comes at little cost w.r.t. 2D BB localization and viewpoint estimation, and then move on to the more challenging task of ultra-wide baseline matching, which is only enabled by enforcing across-viewpoint constraints.

2D bounding box localization. In Tab. 3.3 (up, last col.), we observe a noticeable performance drop from DPM-VOC to DPM-3D-Constraints for both Pascal cars and bicycles for *synthetic* (from 34.5% to 24.9% and 25.2% to 20.7% AP, respectively). Interestingly, this drop is almost entirely compensated by *mixed*, leaving us with remarkable 63.1% AP for cars and 56.8% AP for bicycles, close to the state-of-the-art results (DPM-Hinge). Tab. 3.3 (down, last col.) confirms this result for 3D Object Classes. DPM-3D-Constraints obtains 0.2% lower AP for cars and 2.6% lower AP for bicycles, maintaining performance on par with the state-of-the-art.

Viewpoint estimation. The switch from DPM-VOC+VP to DPM-3D-Constraints results in performance drop, which we attribute to the reduced number of parameters due to the additional 3D constraints. Still, this performance drop is rather small. In particular for *mixed* (we lose only 1.3% MPPE for cars and 2.5% for bicycles).

Ultra-wide baseline matching. In this experiment, we quantify the ability of the DPM-3D-Constraints to hypothesize part positions that are consistent across viewpoints. We adapt the experimental setting proposed by (Zia *et al.*, 2011), and use corresponding part positions on two views of the same object as inputs to a structure-from-motion (SfM) algorithm. We then measure the Sampson error (Hartley and Zisserman, 2004) of the resulting fundamental matrix (see Fig. 3.4), using ground truth correspondences. We use the same subset of 3D Object Classes cars as (Zia *et al.*, 2011), yielding 134 image pairs, each depicting the same object from different views, against static background. Tab. 3.4 gives the results (percentage of estimated fundamental matrices with a Sampson error < 20 pixels), comparing a simple baseline using SIFT point matches (col. 1) to the results by (Zia *et al.*, 2011) (col. 2), and the DPM-3D-Constraints using 12 (col. 3) and 20 parts (col. 4), respectively, for varying angular baselines between views. As expected, the SIFT baseline fails for views with larger baselines than 45°, since the appearance of point features changes too much to provide matches. On the other hand, we observe competitive performance of our 20 part DPM-3D-Constraints compared to (Zia *et al.*, 2011) for baselines between 45° and 135°, and a significant improvement of 29.4% for the widest baseline (180°), which we attribute to the ability of our DPM-3D-Constraints to robustly distinguish between opposite view points, while (Zia *et al.*, 2011) reports confusion for those cases.

Summary. Our results confirm that the DPM-3D-Constraints provides robust estimates of part positions that are consistent across viewpoints, and hence lend themselves to 3D geometric reasoning. At the same time, the DPM-3D-Constraints maintains performance on par with state-of-the-art for both 2D BB localization and viewpoint estimation.

Azimuth	SIFT	Zia *et al.*	DPM-3D-Constr. 12	DPM-3D-Constr. 20
45 °	2.0%	55.0%	49.1%	54.7%
90 °	0.0%	60.0%	42.9%	51.4%
135 °	0.0%	52.0%	55.2%	51.7%
180 °	0.0%	41.0%	52.9%	70.6%
AVG	0.5%	52.0%	50.0%	57.1%

Table 3.4: Ultra-wide baseline matching performance, measured as fraction of correctly estimated fundamental matrices. Results for DPM-3D-Constr. with 12 and 20 parts versus state-of-the-art.

Figure 3.4: Example ultra-wide baseline matching (Zia *et al.*, 2011) output. Estimated epipoles and epipolar lines (colors correspond) for image pairs.

3.5 CONCLUSION

In this chapter, we have presented the first step towards building powerful 3D object representations, contributing with robust viewpoint estimation and 3D part localization methods. In particular, we have shown how to teach 3D geometry to the deformable parts model, aiming at narrowing the representational gap between state-of-the-art object class detection and scene-level, 3D geometric reasoning. By adding geometric information on three different levels, we improved performance over the original DPM and prior work. We achieved improvements for 2D bounding box localization, viewpoint estimation, and ultra-wide baseline matching, confirming the ability of our model to deliver more expressive hypotheses w.r.t. 3D geometry than prior work, while maintaining or even increasing state-of-the-art performance in 2D bounding box localization .

3D^{2}PM - 3D DEFORMABLE PART MODELS 4

Contents

Objects are inherently three-dimensional. While the previous chapter had set the path towards 3D by introducing multi-view and 3D part representations, this chapter embraces the 3D nature of objects and presents full 3D object representation. In particular, here we introduce a full 3D deformable parts representation capable of providing angular accurate viewpoint estimates due to the continuous viewpoint representation.

The representations presented in this chapter continue where the ones from the previous chapter have stopped. While in Chapter 3 the object model is based on a discrete viewpoint representation and per-viewpoint defined part displacement probabilities, in this chapter parts are fully defined in 3D, including the part displacement distributions, leading to a more compact and natural model. In addition, the 3D object representation in this chapter is equipped with a continuous viewpoint representation, leading to angular-accurate viewpoint estimates. In contrast to previous 3D object representations which have proven hard to match to image evidence (Zia *et al.*, 2013a; Marr and Nishihara, 1978; Lowe, 1987), in an extensive experimental evaluation we demonstrate the excellent detection, viewpoint estimation and wide-baseline matching capabilities of the proposed 3D representation, comparable even to the state-of-the-art 2D representations.

4.1 INTRODUCTION

In the early days, 3D representations of objects and entire scenes were considered the holy grail (Marr and Nishihara, 1978; Brooks, 1981; Pentland, 1986; Lowe, 1987). Being more compact and providing a more faithful approximation of the physical world than 2D image projections, they were deemed more powerful w.r.t. reasoning about individual objects, their interactions in complete scenes, and even functions (Stark *et al.*, 1993; Green *et al.*, 1995). However, despite being rich, these representations could not be reliably matched to real-world imagery. As a consequence, they were largely neglected in favor of 2D representations of object classes based on robust local features and statistical learning techniques (Fergus *et al.*, 2003; Leibe *et al.*, 2008; Dalal and Triggs, 2005; Felzenszwalb *et al.*, 2010).

Recently, researchers have reconsidered the 3D nature of the vision problem in the context of scene understanding. Here, 3D information has shown to be valuable to reduce false detections (Hoiem *et al.*, 2008; Ess *et al.*, 2009; Wojek *et al.*, 2010). This has also fuelled the development of multi-view recognition methods (Thomas *et al.*, 2006; Savarese and Fei-Fei, 2007; Yan *et al.*, 2007; Su *et al.*, 2009; Liebelt and Schmid, 2010; Stark *et al.*, 2010; Zia *et al.*, 2011; Payet and Todorovic, 2011; Glasner *et al.*, 2011; Lopez-Sastre *et al.*, 2011), providing viewpoint estimates as additional cue for scene-level reasoning (Bao and Savarese, 2011). Most approaches, however, are still either limited with respect to the degree of 3D modelling, or can not provide competitive performance in terms of 2D BB localization. In particular, the ability to provide richer object hypotheses than 2D BB in the form of viewpoint estimates is typically connected to significantly sacrificing 2D localization performance in comparison to state-of-the-art object detectors.

In this chapter, we aim to combine the best of both worlds, namely, to leverage performance from one of the most powerful 2D object class detectors to date, and a 3D object class representation that allows for fine-grained 3D object and scene reasoning. In this way, we hope to benefit from the compact and rich 3D representation while retaining the robustness in matching to real-world images.

We make several contributions. First, we propose a 3D version of the powerful deformable parts model, DPM, combining the representational power of 3D modelling with robust matching to real-world images. While DPM-VOC+VP and DPM-3D-Constraints (see Chapter 3) are still multi-view object detectors, in this chapter we establish a new, fully 3D parameterized DPM. Second, we demonstrate that our model delivers richer object hypotheses than 2D BB, in the form of viewpoint estimates of arbitrary granularity, and part localization consistent across viewpoints, outperforming prior work. Third, in contrast to previous work on 3D object models, we show competitive performance to state-of-the-art techniques for BB localization.

4.2 EXTENDING DPM-3D-CONSTRAINTS TO 3D

This section introduces our model, a part based model that is a conditional distribution over 3D parts. It can be seen as a 3D version of the DPM (Felzenszwalb *et al.*, 2010). Although the standard DPM has proven successful for object detection, it is ignorant to 3D object geometry. By encoding the underlying 3D object structure we obtain a compact model with a smaller number of parameters. At the same time, we hope to obtain a model that is more descriptive of the 3D object itself.

We describe our model successively. In Sec. 4.2.1 we explain the conditional random field (CRF) model and fix notation. Sec. 4.2.2 describes the pairwise terms of the CRF that are drawing on the 3D part displacement model. Sec. 4.2.3 introduces three versions of the unary term. We propose a discrete model and two continuous ones. Model learning is introduced in Sec. 4.2.4 and inference in Sec. 4.2.5. The final model allows for arbitrary fine viewpoint estimation because of the 3D part displacement and continuous appearance model. In the experiments we carefully analyze the contributions of the individual modeling components.

This model (3D^{2}PM) is strongly related to the DPM-3D-Constraints model (Section 3.3.2). Although DPM-3D-Constraints also uses a structured SVM objective to jointly optimize for detection and viewpoint estimation and includes 3D constraints across viewpoints, the resulting model still is a collection of 2D DPM models, in other words it is a multi- view object detector. As a result it can infer the viewpoint in a discrete set that has to be specified already during training time. The model described here however results in a full 3D model with a continuous appearance model, allowing for arbitrarily fine viewpoint estimation at test- time. Also, the number of parameters to be learned in DPM-3D-Constraints is far larger than for 3D^{2}PM, as the latter one attains more compact part representation.

4.2.1 Preliminaries

Training data contains tuples $\{I_i, y_i\}_{i=1,\dots,N}$, with I the image and y the output variable consisting of three parts $y = (y^l, y^v, y^b) \in \mathcal{Y}$: y^b specifying the bounding box given by its upper, lower, left and right boundary; $y^v \in [0, 360)$ denoting the viewing angle; $y^l \in \{-1, 1, 2, \dots, C\}$ denoting the class membership of the object (-1 in case no object of interest is present). We use a star shaped conditional random field (CRF) to model the dependency of the output variable y on image evidence I. We have a collection of $M+1$ parts, $p_0, p_1, \dots, p_M$, where $p_0 = (x_0, y_0, z_0)$ denotes the observed root part and the remaining parts are latent. Their values $p_i = (x_i, y_i, z_i)$ are defined relative to the root 3D position p_0. Thus, the conditional distribution for a given image I and model parameters β and the latent space denoted by $h = (p_1, \dots, p_M)$ has the following form:

$$p(y, h \mid I, \beta) = Z(I, \beta)^{-1} \exp\left(- \sum_{i=0}^{M} \langle \beta_i^u, \psi^u(I, y, p_i) \rangle - \sum_{i=1}^{M} \langle \beta_i^p, \psi^p(I, p_0, p_i) \rangle \right) \tag{4.1}$$

where $\beta_i^u \in \mathbb{R}^D, i = 0, \ldots, M$ are parameters of the unary term, $\beta_i^p \in \mathbb{R}^{D_p}, i = 1, \ldots, M$ are parameters of pairwise terms, and the ψ^u, ψ^p are the unary and pairwise feature functions, that we specify in the next two sections. For a more compact notation we use β for stacked unary and pairwise terms.

4.2.2 Three-dimensional displacement model

The pairwise terms in equation (4.1) are related to the displacement of part p_i w.r.t so called anchor part positions $v_i = (v_{ix}, v_{iy}, v_{iz})$, which are defined w.r.t. the root part p_0. These displacements are defined as 3D Gaussians with diagonal covariance matrix Σ_i:

$$p(p_i|p_0, v_i) \propto \exp\left(-\frac{1}{2}\left((x_i, y_i, z_i)^\top - (v_i + \mu_i)\right)^\top \Sigma_i^{-1}\left((x_i, y_i, z_i)^\top - (v_i + \mu_i)\right)\right) \quad (4.2)$$

While the anchors are fixed during initialization they allow for free movement of the parts in 3D. To obtain the pairwise terms ψ^p the 3D part displacement distribution is projected onto a particular viewpoint (see Fig. 4.1). As a general perspective projection can result in a non-Gaussian distribution we restrict ourselves to a scaled orthographic projection Q^v instead. While this is clearly an approximation it works well in practice in particular when the object is relatively far way. To get more accurate approximation, we introduce a separate scaled orthographic projection Q_i^v for each part p_i. Each Q_i^v has a unique scaling factor related to the depth of the anchor of p_i in this particular view. For a given part p_i the projected 2D part displacement distribution has a mean $\mu_i^v = Q_i^v \mu_i$ and covariance $\Sigma_i^v = (Q_i^v)\Sigma_i(Q_i^v)^T$. As Q_i^v is a linear transformation, the resulting 2D part displacement distribution remains a Gaussian distribution. This distribution is fully parameterized with 6 parameters per part. This is in contrast to DPM-3D-Constraints where separate displacement models are trained for K different viewpoints resulting in $4K$ parameters. Thus this model is compact but also comes with fewer degrees of freedom.

Going from 3D to 2D, each part $p_i = (x_i, y_i, z_i)$ in 2D is parameterized as $p_i^v = (u_i, v_i, l_i)$, where $(u_i, v_i) = Q_i^v p_i$ and l_i is the resolution in image space and is typically fixed for each part to be at twice the resolution of the root filter (Felzenszwalb *et al.*, 2010). The root itself $p_0 = (u_0, v_0, l_0)$ has a (u_0, v_0) position in the 2D image and a resolution parameter l_0. Recall the output variables y^l, y^v, y^b. There is a deterministic relationship between the position and resolution of the root filter p_0 and the bounding box of the training example, which is why we speak of p_0 and y^b interchangeably. Finally, the pairwise term is defined as follows, closely following the original DPM formulation (Felzenszwalb *et al.*, 2010):

$$\begin{aligned}\left\langle \beta_i^p, \psi^p(I, p_0, p_i) \right\rangle = \\ \left\langle \left((\Sigma_i^v)_{11}^{-1}, (\mu_i^v)_1, (\Sigma_i^v)_{22}^{-1}, (\mu_i^v)_2, (\Sigma_i^v)_{12}^{-1}\right), \left(-du^2, -du, -dv^2, -dv, -2dudv\right) \right\rangle\end{aligned} \quad (4.3)$$

where (du, dv) are the offsets of the projected part from the projected anchor, measured in the image plane.

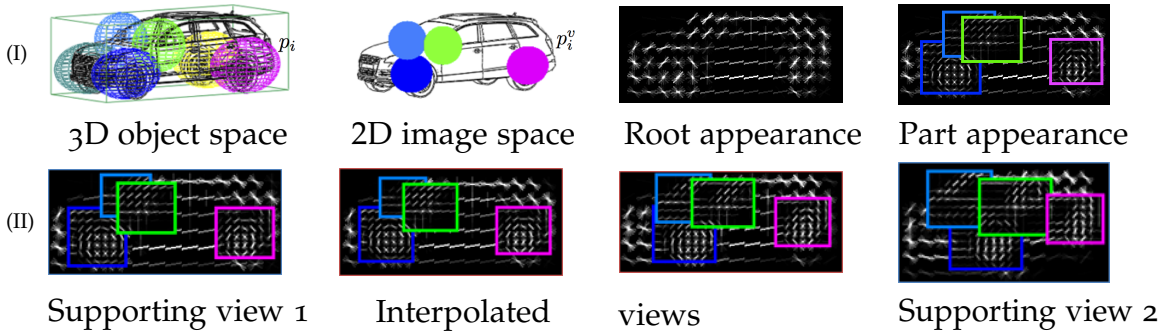


Figure 4.1: Part displacement distributions and continuous appearance model. (I) Left to right: Learned 3D part displacement distributions, part projections in an arbitrary view (some 3D parts not visible due to occlusion), root and part appearances at the given view. (II) Continuous appearance model. First and last column: two supporting views, middle: two interpolated views.

4.2.3 Continuous appearance representation

Although our goal is to have a full 3D object model with continuous appearance representation, we start by introducing a discrete appearance model and then describe two continuous versions. We divide the viewing circle $[0, 360)$ into K different bins. Conceptually the easiest model is to train K different filters, one for each bin and then use this filter as a unary factor for all those y where y^v falls into the bin. We are going to refer to this model as 3D^{2}PM-D, where "D" stands for discrete appearance bins. Thus the factor ψ_k^u for an appearance bin k is represented through the root factor $\psi_{0,k}^u$ and the part factors $\psi_{i,k}^u$ for that particular bin. We use HOG (Dalal and Triggs, 2005) as features. The unary terms for the different parts are given by:

$$\langle \beta_i^u, \psi^u(I, y, p_i) \rangle = \sum_{k=1}^{K} \mathbf{1}_{y^v \in k} \langle \beta_{i,k}^u, \psi_{i,k}(I, y, h) \rangle. \tag{4.4}$$

In order to arrive at a continuous viewpoint model we need to specify unary potentials for arbitrary viewpoint y^v beyond the K bins. For this we interpolate among the unary filters of the appearance bins (called 3D^{2}PM-C in the following, see Fig. 4.1). The continuous models allow for establishing arbitrarily fine viewpoint estimation as the appearance is not restricted to a set of K bins. Note that in this case there are no actual bins as we do not perform binning but rather use the appearance in so called supporting viewpoints among which the continuous appearance model interpolates. Only for naming consistency with 3D^{2}PM-D, we will refer to the supporting viewpoints as bins. We explore two interpolation schemes, namely linear interpolation and exponential interpolation. In the linear interpolation scheme, the continuous appearance is defined as a linear combination of the appearance in the

discrete appearance bins

$$\langle \beta_i^u, \psi^u(I, y, p_i) \rangle = \sum_{k=1}^{K} \alpha_k \langle \beta_{i,k}^u, \psi_{i,k}(I, y, h) \rangle \tag{4.5}$$

where $\alpha_k \propto \angle(y^v, y_k^v)$ is proportional to the angular distance between the viewpoint y^v of the example and the viewpoint of the k-th appearance bin y_k^v. In the exponential interpolation scheme we assign exponential weighting to the unaries of the appearance bins.

$$\langle \beta_i^u, \psi^u(I, y, p_i) \rangle = \sum_{k=1}^{K} e^{-d^2(y^v, y_k^v)} \langle \beta_{i,k}^u, \psi_{i,k}(I, y, h) \rangle \tag{4.6}$$

where $d(y^v, y_k^v) \propto \angle(y^v, y_k^v)$. In the experiments described below we analyze and compare all three models in terms of detection performance and viewpoint estimation accuracy.

4.2.4 Model learning

We train our models using a latent variable structured SVM objective with margin-rescaling (Yu and Joachims, 2009)

$$\begin{aligned} \min_{\beta^u, \beta^p \geq 0, \xi \geq 0} \quad & \frac{1}{2}\|\beta\|^2 + \frac{C}{N}\sum_{n=1}^{N} \xi_n \\ \text{sb.t.} \quad & \max_{h_n} \langle \beta, \psi(I_n, y_n, h_n) \rangle - \max_{\bar{h}} \langle \beta, \psi(I_n, \bar{y}, \bar{h}) \rangle \geq \Delta(y_n, \bar{y}) - \xi_n, \forall \bar{y} \in \mathcal{Y}. \end{aligned} \tag{4.7}$$

The loss function Δ is task dependent and as we are interested in object localization and accurate viewpoint estimation, in this work we use a linear combination of an object detection error and a term measuring viewpoint estimation accuracy

$$\Delta(y, \bar{y}) = \alpha \Delta_{VOC}(y, \bar{y}) + (1 - \alpha)\Delta_{vp}(y, \bar{y}). \tag{4.8}$$

Here Δ_{VOC} is the object detection error. We use the same form of Δ_{VOC} as in (Blaschko and Lampert, 2008) which penalizes detections with insufficient ground truth overlap. The viewpoint estimation loss Δ_{vp} can attain two forms, depending if we are interested into performing viewpoint classification or angular viewpoint estimation. In the case of viewpoint classification, we use $0/1$ loss and in the case of angular viewpoint estimation we use angular precise loss $\Delta_{vp}(y, \bar{y}) = \frac{\angle(y^v, \bar{y}^v)}{180^\circ}$. 3D^{2}PM-D uses the $0/1$ viewpoint loss and thus optimizes for viewpoint classification and 3D^{2}PM-C uses the continuous viewpoint loss and thus optimizes for angular precise viewpoint estimation. We use $\alpha = 0.5$ in the experiments.

The learning leverages on 3D information from CAD data. Following (Stark *et al.*, 2010) we use wireframe-like non-photorealistic rendered images of CAD models.

CAD data is used for unsupervised part initialization as well. We perform the same part initialization as DPM but in 3D. First, a 3D grid of possible part placements is defined. The parts have predefined size (sx, sy, sz) and each candidate part location gets an appearance energy score, which is a sum of the appearances of the corresponding projected parts across views. From this set, the top k parts, which attain the highest appearance score are chosen.

Training. We employ stochastic gradient descend to solve the non convex max-margin optimization problem. We use an expectation-maximization like technique where we iterate between inferring the latent variables h, while the model β is fixed and training the model β for fixed latent variables (the non-root parts), similar in spirit to the one presented in Chapter 3, except that in the 3D^{2}PM case we directly learn the 3D parameters through the max-margin framework (see Eq. 4.7).

4.2.5 Inference

Finally, test time inference solves the following problem $\text{argmax}_{y,h}\langle\beta, \psi(I, y, h)\rangle$ and can be computed using the max-product algorithm. This will infer the position of an object, its viewpoint and also all parts. The inference problems for 3D^{2}PM-D and 3D^{2}PM-C differ. In the case of 3D^{2}PM-D, every test example gets assigned to one of the K appearance bins (viewpoint classes) also determining it's viewpoint label. As in (Felzenszwalb *et al.*, 2010) parts are inferred using efficient distance transform. The maximum scoring bin determines the viewpoint of the test example.

In the case of 3D^{2}PM-C viewpoint inference is a continuous problem due to the continuous nature of the appearance model. In practice we resort to establishing inference on an arbitrarily fine viewpoint resolution (obtained by interpolation) as enabled by the continuous nature of the appearance model. Note that the decision at which viewpoint resolution the model is evaluated is done during test-time and not before training the model. In the experiments we will report on the accuracy for viewpoint estimation depending on the number of appearance bins used during training.

4.3 EXPERIMENTS

In this section we thoroughly evaluate our model, by successively adding 3D information, going beyond plain 2D bounding box localization. While gradually going towards full 3D object model, we first consider the task of coarse viewpoint estimation, where we compare 3D^{2}PM-D and 3D^{2}PM-C models (Sect. 4.3.1). In a second step, and different from previous work in multi-view recognition, we aim at providing arbitrarily fine viewpoint estimation in real world images by leveraging on the full 3D nature of our 3D^{2}PM-C model (Sect. 4.3.2 and 4.3.3). While improving state-of-the-art results on standard benchmarks for fine viewpoint estimation, we

give a detailed analysis of different aspects of 3D^{2}PM-C and describe an coarse to fine viewpoint estimation inference (Sect. 4.3.5).

Even-though the focus is on viewpoint estimation, we realize that a viewpoint estimation system has to be performant on the task of object detection as well, and hence provide object detection results for all viewpoint benchmarks as well as on the challenging Pascal VOC 2007 (Everingham *et al.*, 2007) dataset, where we show superior detection performance to all previous 3D object models in the literature.

We further compare different data sources for training, namely, real world images and synthetic images in the form of rendered CAD models, motivated by previous work leveraging CAD models (Liebelt and Schmid, 2010; Stark *et al.*, 2010). We show that synthetic data can improve viewpoint estimation due its ability to provide perfect annotation despite its appearance statistics that differs from real images. We use 41 commercial cars (www.doschdesign.com) and 43 bicycles (www.sketchup.google.com).

4.3.1 Coarse-grained viewpoint estimation

We start by evaluating the 3D^{2}PM-D on coarse viewpoint estimation, phrased as a multi-class classification problem (viewpoint binning). We report results for cars and bicycles of 3D Object classes (Savarese and Fei-Fei, 2007), a challenging benchmark data set tailored towards multi-view recognition (8 different viewpoint bins). In all experiments, we train from real images provided by the respective dataset as well as CAD data, which serve as a 3D proxy for our model and provide natural 3D constraints across different views of the same instance. We follow the testing protocols of Savarese and Fei-Fei (2007); Everingham *et al.* (2007); Lopez-Sastre *et al.* (2011) and report Mean Precision of Pose Estimation (MPPE) (Lopez-Sastre *et al.*, 2011) as a measure of viewpoint classification accuracy (diagonal average of confusion matrix). We evaluate detection performance using Average Precision (AP) as established in the Pascal VOC (Everingham *et al.*, 2007) challenge.

Results. Tab. 4.1 shows results, including 3D^{2}PM-D (last col.), recent successful 3D object models (second row), and various 2D models (first row), notably the state-of-the-art multi-view object detector DPM-VOC+VP (Chapter 3). We observe 3D^{2}PM-D to achieve 95.8% and 96.0% MPPE on cars and bicycles, respectively, outperforming all previous work using 3D object models (85.3% cars (Glasner *et al.*, 2011) and 75.5% bicycles (Liebelt and Schmid, 2010)). Comparing to 2D models 3D^{2}PM-D performs on par with DPM-3D-Constraints (96.3% and 96.4%, Col. 3) and it is slightly worse than DPM-VOC+VP (97.9%, 98.9%). The object detection results show similar tendency. 3D^{2}PM-D with 99.6% AP and 94.1% AP on cars and bicycles outperforms all previous work using 3D models (99.2% and 69.8%), it performs on par with DPM-3D-Constraints (99.7% and 95.0%), and is slightly worse than DPM-VOC+VP (99.9% with 97.6%).

We stress that 3D^{2}PM-D is trained with full 3D part displacement model across appearance bins. It shows remarkable viewpoint estimation and 2D detection performance on this dataset in comparison to the DPM-VOC+VP model, which is a

	2D Models			
AP/MPPE	Payet and Todorovic	Lopez-Sastre *et al.*	DPM-3D-Constr.	DPM-VOC+VP
cars	- / 86.1	96.0 / 89.0	99.7 / 96.3	99.9 / 97.9
bicycles	- / 80.8	91.0 / 90.0	95.0 / 96.4	97.6 / 98.9

	3D Models			
AP/MPPE	Liebelt and Schmid	Zia *et al.*	Glasner *et al.*	3D^{2}PM-D
cars	76.7 / 70.0	90.4 / 84.0	99.2 / 85.3	99.6 / 95.8
bicycles	69.8 / 75.5	- / -	- / -	94.1 / 96.0

Table 4.1: Viewpoint estimation (in MPPE, Lopez-Sastre *et al.* (2011)) and object detection (in AP) results on car and bicycle class from 3D Object classes (Savarese and Fei-Fei, 2007) dataset.

2D model and directly optimizes for the given task. On the other hand, even though DPM-3D-Constraints is a more complex model than 3D^{2}PM-D, as it models part displacement independently in each view, it shows comparable performance with our 3D^{2}PM-D model.

Summary. In conclusion, 3D^{2}PM-D outperforms previous 3D models and achieves competitive performance to the state-of-the-art multi-view 2D object detectors (Lopez-Sastre *et al.*, 2011), DPM-VOC+VP and DPM-3D-Constraints, despite being less complex due to its full 3D representation.

4.3.2 Fine-grained viewpoint estimation

In a next round of experiments, we go one step further and evaluate 3D^{2}PM-D and 3D^{2}PM-C w.r.t fine-grained viewpoint estimation. To this end, we use EPFL Multi-view cars (Ozuysal *et al.*, 2009) due to the more fine-grained annotations. The data set contains 20 sequences of cars imaged from a full circle of 360 degrees. Angular viewpoint annotations are approximate. We follow Ozuysal *et al.* (2009) and use the first 10 sequences for training and test on the other 10. Viewpoint estimation is again phrased as multiclass classification, but we now vary the granularity of viewpoint sampling. Thus now we have models with k bins for $k \in \{8, 12, 16, 18, 36\}$. In each model, the bin centers have equi-distant spacing of $\frac{360}{k}$. As the annotations are continuous, we evaluate the 3D^{2}PM-C models with linear (3D^{2}PM-C-Lin) and exponential (3D^{2}PM-C-Exp) appearance interpolation as well.

Results. Table 4.2 compares object detection and viewpoint classification performance of our 3D^{2}PM-D and 3D^{2}PM-C models with linear and exponential interpolation to previously published results. For viewpoint estimation, 3D^{2}PM-D with 8 appearance bins achieves 78.5% MPPE which is 5% better than the state-of-the-art result 73.7% MPPE of Lopez-Sastre *et al.* (2011). 3D^{2}PM-C-Lin and 3D^{2}PM-C-Exp achieve comparable accuracy of 78.3% and 77.9%, respectively, also improving over

AP/MPPE	Ozuysal *et al.*	Lopez-Sastre *et al.*	3D^{2}PM-D	3D^{2}PM-C-Lin	3D^{2}PM-C-Exp
8 bins	- / -	91.0 / 73.7	99.4 / 78.5	97.8 / 78.3	98.1 / 77.9
12 bins	- / -	- / -	97.9 / 75.5	98.3 / 76.2	98.4 / 77.3
16 bins	85.0 / 41.6	97.0 / 66.0	99.0 / 69.8	97.5 / 69.0	98.0 / 69.1
18 bins	- / -	- / -	99.2 / 71.8	99.3 / 71.2	99.2 / 70.5
36 bins	- / -	- / -	99.3 / 45.8	99.2 / 52.1	99.5 / 53.5

Table 4.2: Detection (AP) and viewpoint estimation (MPPE, Lopez-Sastre *et al.* (2011)) (EPFL dataset).

previous work. 3D^{2}PM-D with 16 bins achieves 69.8% MPPE which is by 4% better than the previous state-of-the-art result of Lopez-Sastre *et al.* (2011) and by 28.2% better than Ozuysal *et al.* (2009). 3D^{2}PM-C-Lin and 3D^{2}PM-C-Exp with MPPE of 69% and 69.1%, respectively, similarly outperform previous work and are on par with 3D^{2}PM-D. In terms of detection, 3D^{2}PM-C-Lin and 3D^{2}PM-C-Exp with 8 bins achieve 97.8% AP and 98.1% AP, outperforming the 91.0% of Lopez-Sastre *et al.* (2011), while 3D^{2}PM-D achieves 99.4% AP which is in the range of the 3D^{2}PM-C models. For 16 bins, 3D^{2}PM-D, 3D^{2}PM-C-Lin and 3D^{2}PM-C-Exp achieve 99%, 97.5% and 98% AP and collectively outperform the state-of-the-art results 97% of Lopez-Sastre *et al.* (2011) and 85% (Ozuysal *et al.*, 2009). Increasing viewpoint granularity from 8 to 36 bins, we observe that detection performance stays roughly the same in the range of 98-99% AP for all 3D^{2}PM variants. For viewpoint classification, performance seems to drop. This is in fact an artifact of the MPPE measure accounting only for 0/1 error. Interestingly, the 3D^{2}PM-C-Lin and 3D^{2}PM-C-Exp with 36 appearance bins achieve MPPE of 52.1% and 53.5% and outperform the 45.8% of 3D^{2}PM-D confirming that the continuous appearance model can be more suited for fine viewpoint estimation as it accounts for fine appearance variations.

As EPFL Multi-view cars offer angular viewpoint annotations, we also evaluate the Median Angular Error (MAE) as in Glasner *et al.* (2011), quantifying the more meaningful continuous angular error rather than 0/1 error as it is the case for MPPE.

Tab. 4.3 reports MAE for our 3D^{2}PM models comparing to state-of-the-art. Since the 3D^{2}PM-C uses continuous appearance models, we evaluate it at a finer viewpoint sampling of k bins. This enables us to explore the advantage of having continuous

MAE	Glasner *et al.*	3D^{2}PM-D	3D^{2}PM-C-Lin	3D^{2}PM-C-Exp
8 bins	24.8	12.9	11.1	9.6
12 bins		9.0	7.8	8.8
16 bins		7.2	6.9	7.5
18 bins		6.2	5.6	6.9
36 bins		5.8	4.7	4.7

Table 4.3: Fine viewpoint estimation in MAE (Glasner *et al.*, 2011) (EPFL dataset).

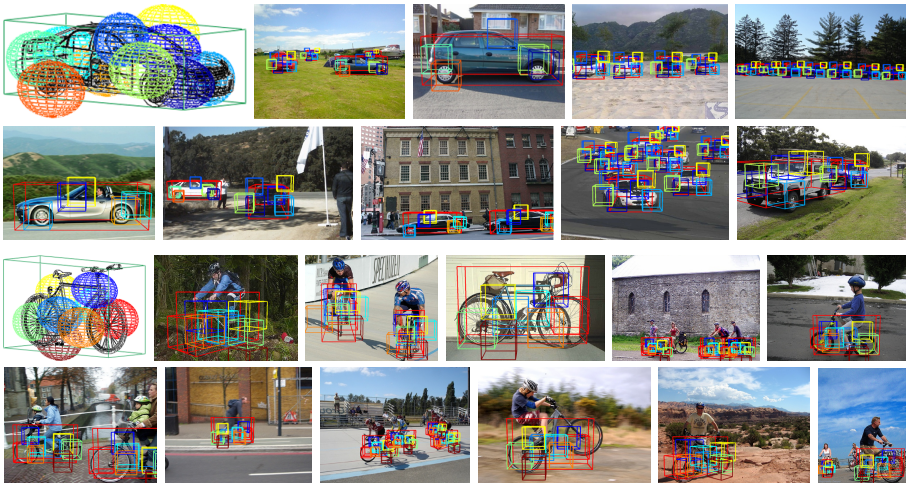


Figure 4.2: Object detection and 3D pose estimation. Example car and bicycle detections on Pascal 2007 (Everingham *et al.*, 2007). Learned part distributions. The 3D part detections are color coded.

appearance modelling in comparison to discrete 2D modelling employed by the rest of the models and all previous work. Our 3D^{2}PM-D, 3D^{2}PM-C-Lin and 3D^{2}PM-C-Exp models with 8 bins achieve 12.9°, 11.1° and 9.6° MAE outperforming by almost 15° the best published result of 24.8° of (Glasner *et al.*, 2011).

Analyzing the different granularities of viewpoint estimation, MAE reduces as we go from coarse (8 bins) to finer viewpoint sampling (36 bins) and the 3D^{2}PM-C models achieve 4.7° MAE with 36 bins which is better than 5.8° of 3D^{2}PM-D, again confirming the intuition that modelling objects in their natural form (in this case the continuous viewpoint appearance) leads to improved performance.

4.3.3 Arbitrarily fine viewpoint estimation

We proceed to evaluate the ability of the 3D^{2}PM-C to generate viewpoint estimates of arbitrary fine granularity, enabled by its continuous appearance representation. We use EPFL Multi-view cars as the only dataset providing angular accurate viewpoint annotation. Our goal is to understand better the 3D^{2}PM-C models and analyze its behavior in different settings. We train 3D^{2}PM-C with k bins, where again $k \in \{8, 12, 16, 18, 36\}$ and try to interpolate from the starting k viewpoints to arbitrarily fine viewpoint resolution. We go dense on the viewpoint sampling as the dataset permits, i.e. up to the label noise of the dataset. We evaluate at viewpoint resolution of 5°, 8°, 10°, 20°, 22.5°, 30°, and 45°.

Results. Fig. 4.3 and Tab. 4.4 give the results for 3D^{2}PM-C-Lin (left) 3D^{2}PM-C-Exp (right). At a coarse level, it is evident that for both models better viewpoint estimation is obtained at finer viewpoint resolution regardless of k. Exploring the other dimension in the plot (number of appearance bins), going from 8 to 36 bins increases performance.

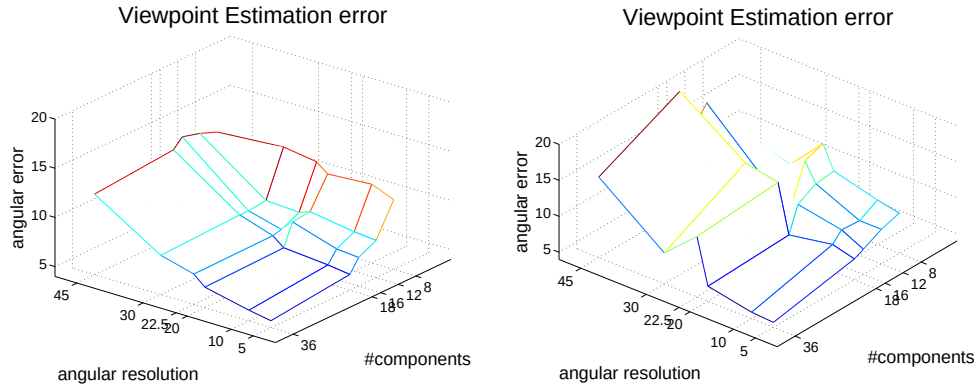


Figure 4.3: Graphical representation of viewpoint classification results, left - linear interpolation, right - exponential. The number of components is the number bins.

Considering the respective best results, the 3D^{2}PM-C-Lin and 3D^{2}PM-C-Exp with 36 bins provide 4.7° MAE, better than any other result reported in the literature approaching the dataset label noise. The 3D^{2}PM-C models with 8 model viewpoints achieve remarkably good viewpoint estimation performance (MAE = 9.6° of 3D^{2}PM-C-Exp and 11.1° of 3D^{2}PM-C-Lin) despite large angular distance between the bins, even on the finest evaluation level of 5° resolution.

Comparing linear vs. exponential appearance interpolation, both models achieve comparable performance on the finer viewpoint resolution levels (5°, 10°, 20°, 22.5°). However, for coarser viewpoint resolution and wider viewpoint spacing among the bins, exponential interpolation provides worse results than linear.

Summary . While the 3D^{2}PM-C model is a simple continuous model, it achieves good performance even when starting from wide angular spacing among the model viewpoint (appearance) bins.

4.3.4 CAD vs. real image data

We want to explore the performance impact of using CAD data, as they have unrealistic appearance but perfect viewpoint annotation. Thus we train models on synthetic data only (synthetic), on real & synthetic (mixed) and on real data where

AP/MAE	at 5°	at 10°	at 20°	at 22.5°	at 30°	at 45°
3D^{2}PM-C-Lin 8	96.8 / 11.1	95.6 / 12.0	96.1 / 11.7	96.9 / 12.6	97.3 / 13.1	97.8 / 12.6
3D^{2}PM-C-Lin 12	98.5 / 7.8	98.6 / 8.0	95.9 / 8.7	97.9 / 8.3	98.3 / 8.5	97.2 / 13.3
3D^{2}PM-C-Lin 16	97.8 / 6.9	98.2 / 7.1	96.8 / 8.5	97.5 / 7.4	97.6 / 8.3	95.0 / 13.8
3D^{2}PM-C-Lin 18	99.1 / 5.6	99.0 / 5.9	99.3 / 6.3	98.6 / 7.2	98.5 / 8.3	97.0 / 12.9
3D^{2}PM-C-Lin 36	99.2 / 4.7	98.8 / 5.1	98.5 / 6.1	98.2 / 7.1	98.0 / 8.0	97.5 / 12.2

Table 4.4: Fine-grained viewpoint estimation in MAE (EPFL dataset).

	3D^{2}PM-D			3D^{2}PM-C		
AP/MAE	mixed	real	synthetic	mixed	real	synthetic
8 bins	99.4 / 12.9	99.2 / 13.1	95.4 / 14.2	97.8 / 12.6	98.3 / 13.7	94.5 / 13.9
18 bins	99.2 / 6.2	99.1 / 7.4	94.9 / 9.8	99.3 / 6.3	98.2 / 7.0	95.6 / 7.1
36 bins	99.3 / 5.8	99.0 / 6.4	95.2 / 7.9	98.8 / 5.1	98.7 / 5.6	96.3 / 5.9

Table 4.5: Real vs. mixed data setting on 3D^{2}PM-C.

we use CAD data only for model initialization. We do experiments with 3D^{2}PM-D and 3D^{2}PM-C models with 8, 18 and 36 bins on the EPFL dataset.

Tab. 4.5 gives the result. Synthetic models with 36 bins achieve very good viewpoint classification performance of 5.9° and 7.9° for 3D^{2}PM-C and 3D^{2}PM-D, respectively, while also achieving good detection results of 96.3% and 95.2% AP. Adding real data (mixed) leads to improved results of 5.8° MAE for 3D^{2}PM-D and 5.1° MAE for 3D^{2}PM-C, while real data only with 6.4° MAE and 5.6° performs worse, speaking in favor of using CAD data with accurate annotation.

4.3.5 Coarse-to-fine viewpoint inference

As we go towards arbitrarily fine viewpoint estimation with 3D^{2}PM-C, we increase the number of model evaluations for a given position and viewpoint (atomic operation). As a result, inference becomes slow. Thus we propose a speed up by using a coarse-to-fine inference that minimizes atomic operations while not sacrificing too much performance. We use a greedy, binary search-like scheme that recursively partitions the space of candidate viewpoints considered.

Results. Tab. 4.6 gives results on EPFL. 3D^{2}PM-C with 36 bins and full inference (row 1) is compared to the same model with coarse-to-fine inference (row 2) starting at 12 viewpoints. The last two rows are 3D^{2}PM-C models trained with 12 and 18 bins, respectively, evaluated on 5°. While achieving almost 5 times faster runtime (0.48×10^{10} vs. 2.2×10^{10} atomic operations), we obtain comparable detection results to the full inference model (99.0% AP and 99.2% AP with coarse-to-fine and full inference) and slightly worse viewpoint estimation results (7.0° vs. 4.7°).

AP / MAE	at 5°	#atomic operations
3D ^{2}PM-C b36 full	99.2 / 4.7	2.20×10^{10}
3D ^{2}PM-C b36 coarse to fine	99.0 / 7.0	0.48×10^{10}
3D^{2}PM b12	97.6 / 7.5	2.20×10^{10}
3D^{2}PM b18	98.0 / 6.9	2.20×10^{10}

Table 4.6: Detection (AP) and vp. estimation (MAE). Full vs. coarse-to-fine inference.

Interestingly, compared to models trained with 12 and 18 appearance bins, we achieve better results while attaining much smaller number of atomic operations. More sophisticated methodologies for search space pruning, such as Branch and Rank (Lehmann *et al.*, 2011), could further improve that trade-off.

4.3.6 Pascal VOC 2007 detection

While previous work on 3D Object models typically reports results on multi-view benchmarks, we evaluate detection performance of the 3D^{2}PM model on the standard detection benchmark Pascal VOC 2007 (Everingham *et al.*, 2007). This is important, since viewpoint estimation is inherently dependent on accurate object localization. Some visual results are shown in Fig. 4.2.

3D^{2}PM-D achieves 61.2% AP on cars, outperforming the previous best 3D Object Model result of 32% AP of (Glasner *et al.*, 2011) by a large margin. Still, DPM-VOC achieves 4% better result of 65.7% AP. DPM-3D-Constraints achieves 63.1%. On bicycles, DPM-VOC and DPM-3D-Constraints achieve 61.3% and 56.8% AP which is better than 3D^{2}PM-D's 52.1% AP. However, given its full 3D nature, 3D^{2}PM's performance is encouraging.

4.3.7 Ultra-wide baseline matching

Lastly, we leverage the 3D nature of our model and the resulting ability to match parts across different viewpoints. We quantify this ability in the form of the ultra-wide baseline matching task established by (Zia *et al.*, 2011).

Tab. 4.7 gives results comparing to pure SIFT matches, (Zia *et al.*, 2011), and DPM-3D-Constraints. 3D^{2}PM-D with 20 parts with 66% of correctly estimated matrices, provides better performance than the DPM-3D-Constraints with 20 parts(54%) and better than 50% of (Zia *et al.*, 2011). We observe a significant improvement of 17.3% to DPM-3D-Constraints 12 and 29.6% to (Zia *et al.*, 2011) on the wide baseline matching task of 180°, which we attribute to the ability of 3D^{2}PM-D to better distinguish opposing views.

Azimuth	SIFT	Zia *et al.*	DPM-3D Const.	DPM-3D Const.	3D^{2}PM-D	3D^{2}PM-D
#corr	-	36	12	20	12	20
45 °	2.0%	55.0%	49.1%	54.7%	47.2%	58.5%
90 °	0.0%	60.0%	42.9%	51.4%	54.3%	77.1%
135 °	0.0%	52.0%	55.2%	51.7%	44.8%	58.6%
180 °	0.0%	41.0%	52.9%	70.6%	70.6%	70.6%
AVG	0.5%	52.0%	50.0%	57.1%	54.2%	66.4%

Table 4.7: Ultra-wide baseline matching performance, measured by the % of correctly estimated fundamental matrices. Second row shows the number of correspondences.

4.4 CONCLUSION

In this chapter, we have presented a 3D object representation which combines features from one of the most powerful object detector to date, the DPM, and a 3D object class representation. Being the first extension of the DPM to a full 3D object model, the 3D^{2}PM leverages on 3D information provided by CAD data, performing viewpoint estimation at arbitrarily fine granularity and achieves state-of-the-art results on viewpoint estimation and wide-baseline matching. At the same time, it performs on par with state-of-the-art 2D object detectors w.r.t. detection performance, while significantly outperforming previous 3D object models. Therefore, the 3D^{2}PM takes a step towards bridging the gap between object detection and higher level tasks like scene understanding and 3D object tracking.

5 MULTI-VIEW AND 3D DEFORMABLE PARTS MODELS

Contents

In the previous two chapters we explored multi-view (Chapter 3) and 3D (Chapter 4) object representations in the context of deformable part models. Since the presented models are structurally related, in this chapter we present a unifying framework for deformable part-based models. Realizing DPMs as star-shaped conditional random fields, here we present our multi-view and 3D DPMs as specific instantiations of the general DPM framework. At the same time, we emphasize the differences across these models and furthermore, we provide further details about the model learning. In addition, we expand the experimental evaluation to several challenging benchmarks. The excellent performance of the 3D DPMs is verified on several object categories in addition to the cars and bicycles considered in the previous chapters.

5.1 INTRODUCTION

Object class detection has reached remarkable performance for a wide variety of object classes, based on the combination of robust local image features with statistical learning techniques (Fergus *et al.*, 2003; Leibe *et al.*, 2004; Felzenszwalb *et al.*, 2010; Girshick *et al.*, 2014; Sermanet *et al.*, 2014). Success is typically measured in terms of 2D bounding box (BB) overlap between hypothesized and ground truth objects (Everingham *et al.*, 2010), favoring algorithms implicitly or explicitly optimizing this criterion (Felzenszwalb *et al.*, 2010).

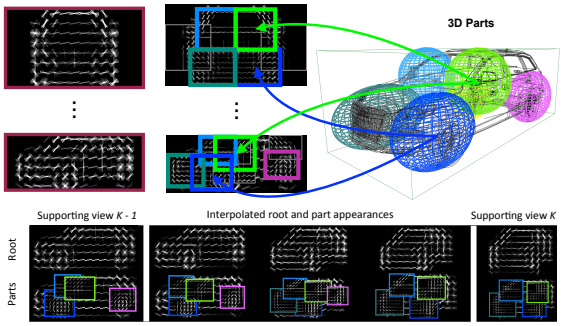


Figure 5.1: 3D^{2}PM model visualization. Learned part 3D displacement distributions along with the continuous appearance model.

Although the state-of-the-art methods for object class detection are appearance based, in the early days of computer vision, geometry based 3D representations of objects and entire scenes were considered the holy grail (Marr and Nishihara, 1978; Brooks, 1981; Pentland, 1986; Lowe, 1987). Being more compact and providing a more faithful approximation of the physical world than 2D image projections, they were deemed more powerful w.r.t. reasoning about individual objects, their interactions in complete scenes, and even functions (Stark *et al.*, 1993; Green *et al.*, 1995). Despite being rich, these representations could not be reliably matched to real-world imagery. As a consequence, they were largely neglected in favor of 2D appearance based representations of object classes. Recently, researchers have reconsidered the 3D nature of the vision problem in the context of scene understanding. Here, 3D information has shown to be valuable to reduce false detections (Hoiem *et al.*, 2008; Ess *et al.*, 2009; Wojek *et al.*, 2010). This has also fueled the development of multi-view recognition methods (Thomas *et al.*, 2006; Savarese and Fei-Fei, 2007; Yan *et al.*, 2007; Su *et al.*, 2009; Liebelt and Schmid, 2010; Stark *et al.*, 2010; Zia *et al.*, 2013a; Payet and Todorovic, 2011; Glasner *et al.*, 2011; Lopez-Sastre *et al.*, 2011), providing richer object hypotheses in the form of viewpoint estimates as additional cue for scene-level reasoning (Bao and Savarese, 2011; Geiger *et al.*, 2011, 2014). However, most approaches are still either limited with respect to the degree of 3D modeling, or can not provide competitive performance in terms of 2D BB localization. In particular, the ability to provide richer object hypotheses than 2D BB is typically associated with sacrificing 2D localization performance in comparison to state-of-the-art object detectors.

In this chapter, we aim to combine the best of both worlds, namely, to leverage performance from one of the most powerful appearance based 2D object class detectors to date, and a geometry based 3D object class representation that allows for fine-grained 3D object and scene reasoning. In this way, we hope to benefit from the natural, compact and rich 3D representation while retaining the robustness in matching to real-world images. The goal is to leave the beaten path towards 2D BB prediction, and to explicitly design an object class detector with outputs amenable to 3D geometric reasoning. By basing our implementation on one of the arguably most

successful 2D BB-based object class detectors to date, the deformable part model (DPM) (Felzenszwalb *et al.*, 2010), we ensure that the added expressiveness of our model comes at minimal loss with respect to its robust matching to real images. To that end, we propose to successively add geometric information to our object class representation, at four different levels.

First, we rephrase the DPM as a genuine structured output prediction task, comprising estimates of both 2D object BB and viewpoint. This enables us to explicitly control the trade-off between accurate 2D BB localization and viewpoint estimation. Second, we introduce 3D geometric constraints on the latent positions of object parts in the DPM. This ensures consistency between parts across viewpoints (i.e., a part in one view corresponds to the exact same physical portion of the object in another view). Third, we extend the notion of discriminatively trained, deformable parts to 3D, by explicitly parametrizing the parts positions and displacement distributions in 3D object coordinates rather than in the image plane (see Fig. 5.1). And fourth, we introduce a continuous appearance representation (see Fig. 5.1), which allows for arbitrarily fine viewpoint estimates in contrast to state-of-the-art multi-view detection methods which can predict only a discrete set of viewpoint classes.

In this chapter, we make the following specific contributions. First, we propose a 3D extension of the powerful DPM, combining the representational power of 3D modeling with robust matching to real-world images. Second, we demonstrate that our models deliver richer object hypotheses than 2D BB, in the form of viewpoint estimates of arbitrary granularity and part localization consistent across viewpoints, outperforming prior work various datasets. Third, in contrast to previous work on 3D object models, we show competitive performance to state-of-the-art techniques for 2D BB localization. Fourth, we use 3D CAD data of the object class of interest mainly as a 3D geometry cue, as well as to enrich the appearance model with rendered images from CAD data. While being not as representative as real world images in terms of feature statistics, these images come with perfect BB and viewpoint annotation, which we can use to improve localization performance and viewpoint estimates.

5.2 MULTI-VIEW AND 3D DEFORMABLE PART MODELS

In this section we introduce our geometry-aware multi-view and 3D object models. We start with the well-known DPM (Felzenszwalb *et al.*, 2010) and gradually introduce 3D geometry cues. This results in a 3D object model, a full 3D version of DPM. The resulting model parameterizes part positions and distributions in 3D and has a continuous appearance representation. We refer to it as $3D^2PM$. Because we encode the underlying 3D object structure, the model becomes more compact with a smaller total number of parameters compared to the DPM. At the same time, we obtain a model that is more descriptive of the 3D object of interest.

We describe our models successively. First, in Sect. 5.2.1 we introduce notation and the idea behind a part-based model. After revisiting the DPM of (Felzenszwalb *et al.*, 2010) in Sect. 5.2.2, we introduce the 2D DPM-VOC+VP in Sect. 5.2.3, a multi-

view object detector which in contrast to the DPM predicts object viewpoint, in addition to the 2D BB. We proceed by introducing 3D geometry into the model and in Sect. 5.2.4 present DPM-3D-Constraints that leverages 3D part constraints, by parameterizing part positions in 3D object coordinates. This establishes part correspondences across different views of the same object. In Sect. 5.2.5 we introduce the 3D object model 3D^{2}PM, parameterizing part positions, as well as displacement distributions in 3D. The 3D^{2}PM includes a continuous appearance model.

5.2.1 Deformable Parts Models as Conditional Random Fields

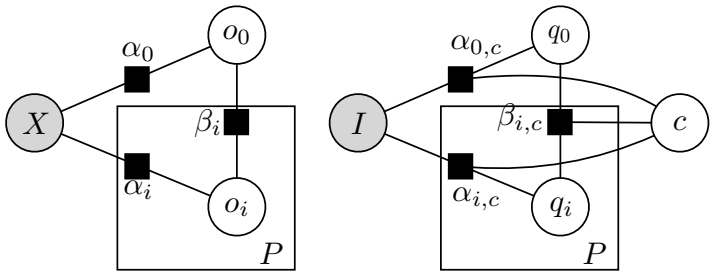
We are given data $\{X\}_{1,\dots,N}$ where X represents an object, defined in image space, or in 3D object coordinates, like a CAD model. The idea behind part-based models is to represent an object by a collection of parts (Felzenszwalb and Huttenlocher, 2005; Felzenszwalb *et al.*, 2010; Andriluka *et al.*, 2009). Previous work has considered different spatial configurations of parts, ranging from star-shaped (Felzenszwalb *et al.*, 2010), tree-shaped (Andriluka *et al.*, 2009), to fully-connected constellations (Stark *et al.*, 2010). Here we build upon the view of a generalized deformable part model as a star-shaped conditional random field (CRF). A star-shaped CRF defines a distribution over object and part positions $\mathbf{o} = (o_0, \dots, o_P)$[5] where o_i denotes an object part, with o_0 being the whole object or the root node and the rest being the child nodes. We define an object part as an axis-aligned hypercube. An object part can be defined in 3D object coordinates as $p_i = [x_1, y_1, z_1, x_2, y_2, z_2]$ in which case the CRF defines a distribution over 3D bounding cubes, or in the image plane as $q_i = [u_1, v_1, u_2, v_2]$, defining a distribution over 2D BBs. Thus, given an object X and a star-shaped CRF model $\boldsymbol{\theta}$, the joint probability distribution over the object hypotheses reads

$$p(\mathbf{o}|\boldsymbol{\theta}, X) \propto \prod_{i=0}^{P} \Psi^u(o_i, \alpha_i, X) \prod_{i=1}^{P} \Psi^p(o_i, o_0, \beta_i). \tag{5.1}$$

This distribution decouples in part-wise terms and for each part o_i there are two terms. First, the unary factor $\Psi^u(o_i, \alpha_i, X)$ scores an object part hypothesis o_i, given the object X. This unary factor is also referred to as the "appearance" term as it captures the appearance of an object part. The second factor is a pairwise term, $\Psi^p(o_i, o_0, \beta_i)$, referred to as the "spatial" term. The pairwise term specifies part o_i placements w.r.t. the root part o_o. All factors are log-linear. We denote the full set of parameters by $\boldsymbol{\theta} = [\boldsymbol{\alpha}, \boldsymbol{\beta}]$ that includes parameters of the unary $\boldsymbol{\alpha} = [\alpha_0, \dots, \alpha_P]$ and pairwise terms $\boldsymbol{\beta} = [\beta_1, \dots, \beta_P]$. For the feature functions we write $\boldsymbol{\phi} = [\phi(o_0, X), \dots, \phi(o_P, X)]$ for the unaries and $\boldsymbol{\eta} = [\eta(o_1, o_0), \dots, \eta(o_p, o_0)]$ for the pairwise features respectively, so the energy of the CRF in this general form reads

$$\langle \boldsymbol{\theta}, \boldsymbol{\psi} \rangle = \langle \boldsymbol{\alpha}, \boldsymbol{\phi} \rangle + \langle \boldsymbol{\beta}, \boldsymbol{\eta} \rangle. \tag{5.2}$$

[5]We use regular font characters to denote part parameters of features. We use characters with bold font whenever we stack parameters from multiple parts or components.



(a) Star-shaped CRF (b) DPM (Felzenszwalb *et al.*, 2010)

Figure 5.2: Graphical models depicting (a) general part-based model as a CRF over the parts o_i conditioned on the data X. (b) the 2D DPM, conditioned on an image I. With shaded nodes, we denote the observed variables.

In the following sections we specify the unary and pairwise terms for each of the models. Fig. 5.2(a) depicts the graphical model defined by the star-shaped CRF.

Previous work on object detection with part-based models (e.g., the DPM (Felzenszwalb *et al.*, 2010)) defines a distribution over 2D object hypotheses by parameterizing parts, unary and pairwise terms in the 2D image space. The models that we present in this chapter extend the DPM, and gradually shift the parameterization from 2D image space to 3D object space, resulting in an object model parameterized entirely in 3D.

5.2.2 DPM-Hinge

The 2D part-based model of Felzenszwalb *et al.* (2010) is one of the most successful object detectors nowadays, as evidenced by its performance on benchmark datasets (Felzenszwalb *et al.*, 2009) and its use as a building block in many subsequent works. Given an image, the DPM outputs a set of 2D BBs, coarsely localizing the objects. In the remainder of this chapter we will refer to the DPM version of Felzenszwalb *et al.* (2010) as DPM-Hinge, as it uses the hinge loss during model learning and allows us to distinguish it from the other models.

Representation. DPM-Hinge is a mixture model with C components, defined in $2D$ image space. Each component $c \in \{1, \dots, C\}$ captures the appearance and part placement of an object in a particular aspect (often coinciding with viewpoint). DPM-Hinge parameterizes an object hypothesis as a collection of 2D BBs of the object q_0 and it's parts $q_1, \dots, q_P$. For an image I, the score of component c for an object hypothesis $\mathbf{q} = [q_0, \dots, q_P]$ is defined as

$$\langle \boldsymbol{\theta}_c, \boldsymbol{\psi}_c(\mathbf{q}, I) \rangle = \sum_{i=0}^{P} \langle \alpha_{i,c}, \phi(q_i, I) \rangle + \sum_{i=1}^{P} \langle \beta_{i,c}, \eta(q_i, q_0) \rangle \tag{5.3}$$

where $\boldsymbol{\theta}_c = [\boldsymbol{\alpha}_c, \boldsymbol{\beta}_c]$ denote unary $\boldsymbol{\alpha}_c$ and pairwise $\boldsymbol{\beta}_c$ parameters of component c. In particular, the parameters collect per-part q_i variables as $\boldsymbol{\alpha}_c = [\alpha_{0,c}, \dots, \alpha_{P,c}]$ and

$\boldsymbol{\beta}_c = [\beta_{1,c}, \ldots, \beta_{P,c}]$. The unary part parameters $\alpha_{i,c}$ are 2D filters for the HOG (Dalal and Triggs, 2005) appearance features $\phi(q_i, I)$. The pairwise factor corresponds to a Gaussian distribution over the part q_i placement relative to q_0. The feature function computes the natural parameters of a 2D Gaussian $\mathcal{N}(q_i|q_0, \mu_{i,c}, \Sigma_{i,c})$. The pairwise features are defined as $\eta_i(q_i, q_0) = -[du_i, dv_i, du_i^2, dv_i^2]$, where $[du_i, dv_i] = q_i - (2q_0 + j_i)$[6]. Here, j_i represents the anchor part position relative to the root. The variables can be understood as $\beta_{i,c} = [\mu_{i,c}^u, \mu_{i,c}^v, \sigma_{i,c}^u, \sigma_{i,c}^v]$, the parameters of a 2D Gaussian.

For the full DPM-Hinge model all parameters from all mixture components are stacked $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_C]$. The graphical model depicting the DPM-Hinge is illustrated in Fig. 5.2(b).

Inference. During inference Felzenszwalb *et al.* (2010) computes the maximum-a-posteriori (MAP) estimate over object hypotheses and components $c^*, \mathbf{q}^* = \text{argmax}_{c,\mathbf{q}} \langle \boldsymbol{\theta}_c, \boldsymbol{\psi}_c(\mathbf{q}, I) \rangle$. This problem involves maximization over two variables, the discrete mixture component c and all part placements $\mathbf{q}$. For each component c the part placement can be found using the efficient distance transform, and the search over c is done by exhaustive enumeration (Felzenszwalb *et al.*, 2010).

Learning. For parameter estimation, the training data is available in pairs $\{(I_i, y_i)\}_{i=1,\ldots,N}$ where I is an image and $y = (y^l, y^b) \in \mathcal{Y}$ is a tuple of annotations. The annotation includes an object class label $y^l \in \{-1, 1, \ldots, L\}$, and a 2D BB y^b.

Felzenszwalb et al (Felzenszwalb *et al.*, 2010) propose to learn the free parameters of their model using a regularized risk objective with the hinge loss. For every object class $k \in \{1, \ldots, L\}$ there is a separate optimization problem

$$\begin{aligned} \min_{\boldsymbol{\theta}, \xi \geq 0} \quad & \frac{1}{2}\|\boldsymbol{\theta}\|^2 + C \sum_{i=1}^{N} \xi_i \\ \text{sb.t.} \quad & \forall i : y_i^l = k : \max_{c, \mathbf{q}_{y_i}} \langle \boldsymbol{\theta}_c, \boldsymbol{\psi}_c(\mathbf{q}_{y_i}, I_i) \rangle \geq 1 - \xi_i \\ & \forall i : y_i^l \neq k : \max_{c, \mathbf{q}_{y_i}} \langle \boldsymbol{\theta}_c, \boldsymbol{\psi}_c(\mathbf{q}_{y_i}, I_i) \rangle \leq -1 + \xi_i. \end{aligned} \tag{5.4}$$

where $\mathbf{q}_{y_i} = [y_i^b, q_1 \ldots, q_P]$, where y^b is the BB of the example and fixed for every training example. The part positions q_i are latent variables, because part annotations are not available. In Felzenszwalb *et al.* (2010) initial values for the component assignments are obtained via aspect ratio clustering and are kept latent during training. This problem is a latent SVM (Felzenszwalb *et al.*, 2010) with hinge loss, which is the reason we refer to the DPM as DPM-Hinge.

5.2.3 DPM-VOC+VP

The DPM-Hinge has shown remarkable performance in terms of 2D object localization, it is however not designed to predict the viewpoint of an object. A multi-view

[6] We use the upper left corners of the parts to compute the displacement features.

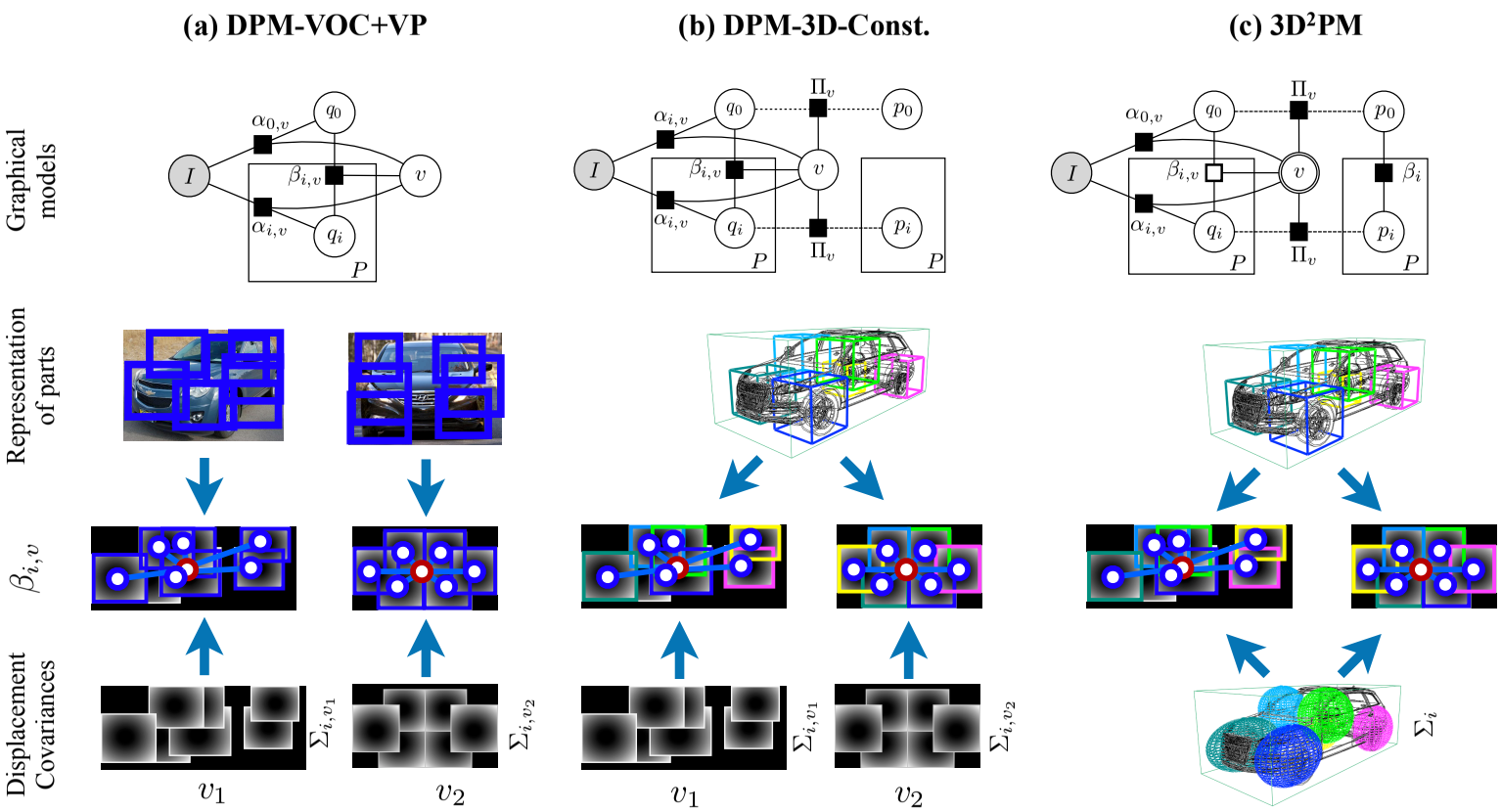


Figure 5.3: Comparison of the different presented models. In the first row from left to right the graphical models of (a) DPM-VOC+VP, (b) DPM-3D-Constraints, and (c) 3D^{2}PM, are shown. In the second row, the part parameterization is illustrated. The third row shows a possible layout of the part configuration. The last row visualizes the covariances of the placement distributions. The variables $\beta_{i,v}$, of the 3D^{2}PM are implicitly defined via projection, see Sect. 5.2.5. Both DPM-3D-Constraints and 3D^{2}PM define parts in a 3D reference frame, therefore it is possible to establish part-correspondences across different viewpoints.

object detector could boost object detection quality and it could be beneficial for high level tasks like 3D scene understanding (Geiger *et al.*, 2011). The first extension we introduce, DPM-VOC+VP, augments DPM-Hinge output with a viewpoint variable v.

Representation. In DPM-VOC+VP we allocate a separate mixture component to each discrete viewpoint v. Every viewpoint component $\boldsymbol{\theta}_v = [\boldsymbol{\alpha}_v, \boldsymbol{\beta}_v]$, has its own unary $\boldsymbol{\alpha}_v = [\alpha_{0,v}, \ldots, \alpha_{P,v}]$ and pairwise $\boldsymbol{\beta}_v = [\beta_{0,v}, \ldots, \beta_{P,v}]$ parameters. DPM-VOC+VP has the same CRF structure as the DPM-Hinge. In addition, it explicitly encodes the object viewpoint v. Fig. 5.3a illustrates the DPM-VOC+VP model.

Inference. The inference is the same as for DPM-Hinge, a MAP estimate over viewpoints and BBs $\mathbf{q}^*, v^* = \operatorname{argmax}_{v,\mathbf{q}} \langle \boldsymbol{\theta}_v, \boldsymbol{\psi}_v(\mathbf{q}, I) \rangle$. We use the same inference technique as DPM-Hinge.

Learning. Since we are interested in joint object 2D localization and viewpoint estimation, we leverage viewpoint annotations in the datasets. We denote the viewpoint class label of a given training example as $y^v \in \{1, \ldots, K\}$, in addition to the BB y^b and the class y^l labels. In contrast to the DPM-Hinge, there is a semantic meaning to the selected component and thus it must be chosen correctly. Therefore we adapt a structured SVM (Yu and Joachims, 2009) with margin rescaling for optimization. This objective has previously been proposed for BB detection in Blaschko and Lampert (2008). The final latent-SSVM optimization problem is

$$\begin{aligned} \min_{\boldsymbol{\theta}, \xi \geq 0} \quad & \frac{1}{2}\|\boldsymbol{\theta}\|^2 + C \sum_{i=1}^{N} \xi_i \\ \text{sb.t.} \quad & \forall i, \bar{y} \neq y_i : \max_{\mathbf{q}_{y_i}} \langle \boldsymbol{\theta}_{y_i^v}, \boldsymbol{\psi}_{y_i^v}(\mathbf{q}_{y_i}, I_i) \rangle \\ & - \max_{\bar{v}, \bar{\mathbf{q}}} \langle \boldsymbol{\theta}_v, \boldsymbol{\psi}_{\bar{v}}(\bar{\mathbf{q}}, I_i) \rangle \geq \Delta(y_i, \bar{y}) - \xi_i \end{aligned} \tag{5.5}$$

where $\mathbf{q}_{y_i} = [y_i^b, q_1, \ldots, q_P]$ as before is the annotated object BB y_i^b with the latent part placements and $\bar{\mathbf{q}}_i = [\bar{y}^b, \bar{q}_1, \ldots, \bar{q}_P]$ a different object hypothesis. Note that for the positive training examples, the viewpoint component is observed. Like in Blaschko and Lampert (2008) we define $\boldsymbol{\psi}_v(\mathbf{q}_{y_i}, I_i) = 0$ whenever $y_i^l = -1$. This has the effect to include the two constraint sets of problem (5.4) into this optimization problem.

The loss function Δ is defined on both the predicted BBs and viewpoint at the same time. We use a convex combination of a BB localization Δ_{VOC} and viewpoint loss Δ_{VP}, namely $\Delta(y, \bar{y}) = \gamma \Delta_{VOC}(y, \bar{y}) + (1 - \gamma)\Delta_{VP}(y, \bar{y})$, with $\gamma \in [0, 1]$.

The performance measure for BB accuracy in the standard benchmarks is the intersection over union score $A(y \cap \bar{y})/A(y \cup \bar{y})$ of two BBs $y, \bar{y}$. Therefore, as proposed in Blaschko and Lampert (2008) we use the following loss function as a proxy

$$\Delta_{\text{VOC}}(y, \bar{y}) = \begin{cases} 0, & \text{if } y^l = \bar{y}^l = -1 \\ 1 - [y^l = \bar{y}^l] \frac{A(y \cap \bar{y})}{A(y \cup \bar{y})}, & \text{otherwise.} \end{cases} \tag{5.6}$$

Algorithm 1: DPM-VOC+VP training algorithm

input : $\{I_i, y_i\}_1^N$ I_i is an image, y_i annotations
output: Trained DPM-VOC+VP $\boldsymbol{\theta}$

1 $\boldsymbol{\theta} \leftarrow$ InitModel(pos, neg)
2 $\mathcal{P} = \emptyset$, $\mathcal{S} = \emptyset$, $\mathcal{N} = \emptyset$
3 **while** *outer loop* **do**
  //Find optimal parts for each positive example
4 **foreach** $i \in pos$ **do**
5 $\mathbf{q}_i \leftarrow \text{argmax}_{\mathbf{q}_i} \langle \boldsymbol{\theta}_{v_i}, \boldsymbol{\psi}_{v_i}(\mathbf{q}_i, I_i) \rangle$
6 $\mathcal{P} = \mathcal{P} \cup [y_i^v, \mathbf{q}_i]$
7 **end**
8 **while** *inner loop* **do**
  //Find a set of violating constraints
9 **foreach** $i \in pos$ **do**
10 $\{[v_i, \bar{\mathbf{q}}_i]\} \leftarrow \text{argmax}_{v,\bar{\mathbf{q}}} \langle \boldsymbol{\theta}_v, \boldsymbol{\psi}_v(\bar{\mathbf{q}}, I_i) \rangle + \Delta([v, \bar{q}_0], y_i)$
11 $\mathcal{S} = \mathcal{S} \cup \{[v_i, \bar{\mathbf{q}}_i]\}$
12 **end**
  //Find a set of hard negative examples
13 **foreach** $i \in neg$ **do**
14 $\{[v_i, \bar{\mathbf{q}}_i]\} \leftarrow \text{argmax}_{v,\bar{\mathbf{q}}_i} \langle \boldsymbol{\theta}_v, \boldsymbol{\psi}_v(\bar{\mathbf{q}}_i, I_i) \rangle$
15 $N = N \cup \{[v_i, \bar{\mathbf{q}}_i]\}$
16 **end**
17 $\boldsymbol{\theta} \leftarrow \text{sgd}(\boldsymbol{\theta}, \mathcal{P}, \mathcal{S}, \mathcal{N})$; //update model
18 **end**
19 **end**

The viewpoint loss Δ_{VP} is the 0/1 classification error with different discrete viewpoint predictions treated as different classes.

In case only the location of the object is of interest, one can set $\gamma = 1$, in which case we refer to the model as DPM-VOC, which uses the same initialization, based on aspect ratio clustering, as for the DPM-Hinge. If both tasks are of interest, we set $\gamma = 0.5$ and refer to the resulting model as DPM-VOC+VP.

Optimization. We solve (5.5) using our own implementation of stochastic gradient descent (SGD) with delayed constraint generation. The latent variables turn the optimization problem into a mixed integer program, solved using coordinate descent. Alg. 1 describes the DPM-VOC+VP learning in detail. We start by initializing the model (line 1), and learning the root appearance terms $\alpha_{0,v}$ for each viewpoint component independently, using a standard SVM. The main part of the algorithm has an outer and inner loop. In the outer loop, the latent parts $\mathbf{q}_i$ are found for every positive training example I_i (line 5), resulting in a set $\mathcal{P}$ of positive training examples. Then in an inner loop, the top K ($K = 10$ in our experiments) active violating constraints $[v_i, \bar{\mathbf{q}}_i]$ are found for every positive training example $\{y_i, I_i\}$.

This is the loss-augmented inference problem (line 10) and yields a current set of active constraints $\mathcal{S}$. We choose the violating constraints such that $\frac{A(\bar{q}_0 \cup y^b)}{A(\bar{q}_0 \cap y^b)} > 0.1$. Then, we search for negative examples from the negative labeled images (line 14), resulting in a set of "hard" negative examples $\mathcal{N}$. Finally the model parameters $\boldsymbol{\theta}$ are found by SGD (line 17).

5.2.4 DPM-3D-Constraints

The DPM-VOC+VP parameterizes part positions in 2D image space, independently across viewpoints. In this section we introduce the DPM-3D-Constraints that fundamentally changes the parameterization and works with parts in 3D. This way of modeling reflects the nature of the problem, observed are only 2D projections of what really are physical objects in a 3D world. Therefore, a parameterization in 3D appears both more meaningful and also should be beneficial for applications such as 3D object tracking or multi-view reconstruction.

Since annotated data is only available as 2D information, we use CAD models of the object classes of interest in addition to the annotated images. Being constructed of triangular surface meshes, 3D CAD models provide geometric descriptions of object class instances, lending themselves to 3D part parameterizations.

Representation. The DPM-3D-Constraints has the same graph structure as the DPM-VOC+VP, (see Fig. 5.3b). The difference is for every discrete viewpoint component there is a perspective projection matrix Π_v that connects the 3D parameterization of parts with the 2D part placement observation (see Fig. 3.2).

For every part p we need to specify the appearance (unary factor), and 2D part placement (pairwise factor). We use the setup of Stark *et al.* (2010) to generate a non-photorealistic, gradient-based renderings of 3D CAD models. The renderings are used to compute HOG features for each part p. From a 3D bounding cube p_i of a part, with $q_i = \Pi_v p_i$[7] we denote the 2D BB obtained by projecting the part into the viewpoint v. Then, the appearance features of the part are computed from the projected BB $\boldsymbol{\psi}(p_i, v, I) := \boldsymbol{\psi}(\Pi_v p_i, I)$.

The pairwise factor acts on 3D parts and computes the relative placement in the projected space. For the 3D root p_0 and part p_i, the feature function of the DPM-VOC+VP is re-used, but after projections $\eta(p_i, p_0, v) = \eta(\Pi_v p_i, \Pi_v p_0)$. There are separate parameters β_v for every viewpoint component v.

In summary, the score of a 3D object hypothesis $\mathbf{p}$ and viewpoint v is

$$\begin{aligned}\langle \boldsymbol{\theta}_v, \boldsymbol{\psi}(\mathbf{p}, v, I)\rangle = &\sum_{i=0}^{P} \langle \alpha_{i,v}, \phi(\Pi_v p_i, I)\rangle + \\ &\sum_{i=1}^{P} \langle \beta_{i,v}, \eta(\Pi_v p_i, \Pi_v p_0)\rangle\end{aligned} \tag{5.7}$$

[7] Although the projection in general results in an arbitrary 2D polygon, we use $q_i = \Pi_v p_i$ to denote the 2D BB surrounding it.

There are two main differences between DPM-3D-Constraints and DPM-VOC+VP. First, the 2D parts q_i are observed as projections Π_v of their 3D counterparts p_i. Second, the model establishes part correspondences between different viewpoints. That is for a CAD model for which multiple renderings from different viewpoints are available, the estimated parts will be in correspondence across the renderings. Fig. 5.3b, illustrates the model. The dotted lines emphasize the deterministic relation (projection Π_v) between the 3D parts p_i and their 2D counterparts q_i.

Inference. The inference problem is the same as for DPM-VOC+VP. We solve for the MAP estimate $\text{argmax}_{v,\mathbf{q}} \langle \boldsymbol{\theta}_v, \boldsymbol{\psi}(\mathbf{q}, v, I) \rangle$. The predicted BB and viewpoint is provided by the highest scoring mixture component.

Learning. The optimization problem, loss function Δ_{VOC+VP}, and algorithm 1 for the DPM-3D-Constraints is the same as for DPM-VOC+VP. Different is the use of CAD data. During learning (Alg. 1, line 5) 3D part positions are inferred over multiple renderings of a CAD model from different viewpoints. We enforce these to be consistent in 3D (thus the name DPM-3D-Constraints).

The training data in the form of images and annotations are augmented with a set of 3D CAD models $\{y^\circ\}$ of the object class of interest. Both are needed. The non-synthetic examples contribute to a realistic appearance model and the CAD models are used to encode 3D object geometry. We found that learning an appearance model from CAD data alone is not expressive enough. Assume we are given a 3D instance y°, then let $S(y^\circ)$ denote the set of all projections of y°. Further we know the precise viewpoint $v_i \; \forall i \in S(y^\circ)$. For a 3D instance the inference is coupled via the set of all its projections

$$p* = \underset{p}{\text{argmax}} \sum_{i \in S(y^o)} \langle \boldsymbol{\theta}_{v_i}, \boldsymbol{\psi}(\Pi_{v_i} p, v_i, I_i) \rangle \,. \tag{5.8}$$

For part initialization, we use the same data-driven method of the DPM-VOC+VP, but now in 3D. First, we define a part to be a 3D cube with size equal to 10% of the largest object size. Second, we choose greedily k non-overlapping part positions with maximal combined appearance score across views.

Training from CAD data allows to implement part-level self-occlusion reasoning effortlessly, using a depth buffer. In each view, we thus limit the number of parts to the ones with visible area higher than 10% of the area of the projected 3D part cube.

5.2.5 3D^{2}PM

In this section we describe the 3D^{2}PM model, a 3D DPM entirely defined in 3D space. The 3D^{2}PM defines a conditional distribution over 3D object hypotheses $\mathbf{p}$ and only implicitly, through marginalization, for 2D object hypotheses $\mathbf{q}$. While DPM-3D-Constraints uses a 3D part parameterization, it is still a mixture model with different mixture components for different viewpoints, being limited to discrete set of viewpoints. The 3D^{2}PM model is continuous in the viewpoint variable $v \in \mathcal{V}$.

Representation. Starting from DPM-3D-Constraints, two ingredients are needed to obtain a full 3D object model: a continuous appearance model, and

a 3D part displacement distribution.

For the definition of the continuous unary factor we introduce a number of support views $v_k, k = 1, \ldots, K$. For a given viewpoint v we then define the unary factor to be the weighted combination

$$\langle \alpha_{i,v}, \phi(\Pi_v p_i, I) \rangle = \sum_{k=1}^{K} w_k(v) \langle \alpha_{i,v_k}, \phi(\Pi_k p_i, I) \rangle \tag{5.9}$$

with $w_k(v)$ being a viewpoint dependent scalar. The parameters of this model are thus the collection of all unary factors for parts and support views $\alpha_{i,v_k}, i = 1, \ldots, P, k = 1, \ldots, K$. In practice, we choose the support views to be equally spaced in angular distance $\delta_v = \angle(v_k, v_{k-1})$ on the viewing circle. This appearance score interpolates for a viewpoint v filters from neighboring viewpoints. We experiment with three different models that correspond to different interpolations (i.e., choices of w_k): (i) linear interpolation, (ii) exponential interpolation, and (iii) a discrete set of views. In (i) we set $w_k = 1 - \frac{\angle(v,v_k)}{\angle(v_{k-1},v_k)}$ for the two closest support views, and $w_k = 0$ for the rest. We refer to this model as 3D^{2}PM-C-Lin, as it uses linear interpolation scheme. In (ii) we set $w_k = \exp(-\angle^2(v, v_k))$ and refer to the model as 3D^{2}PM-C-Exp. Finally, in (iii) we set $w_k = \mathbf{1}_{v=v_k}$ and we refer to this model as 3D^{2}PM-D as it can output a discrete set of viewpoints only.

For a given part p_i and root p_0, the pairwise factor scores the joint displacement, again using a Gaussian term, but different to previous models, in 3D $\langle \beta_i, \eta_i(p_0, p_i) \rangle \propto -\ln(\mathcal{N}(p_i|p_0, \mu_i, \Sigma_i))$. The pairwise parameters are the $\beta_i(p_0, p_i) = [\mu_{ix}, \mu_{iy}, \mu_{iz}, \sigma_{ix}, \sigma_{iy}, \sigma_{iz}]$ and the feature function computes $\eta_i(p_0, p_i) = -[dx, dy, dz, dx^2, dy^2, dz^2]$. This factor contains only six parameters per part, in contrast to the previous models where $4K$ displacement parameters per part are required.

To define the score for a 2D object hypothesis $\mathbf{q}$ in an arbitrary viewpoint v, the 3D part displacement distribution is projected to 2D. For an arbitrary viewpoint v the 3D part displacement distribution is projected via a scaled orthographic projection $Q_{i,v}$. The resulting distribution is the marginal of the Gaussian under this projection. Therefore the mean $\mu_{i,v} = Q_{i,v}\mu_i$, and covariance $\Sigma_{i,v} = Q_{i,v}\Sigma_i Q_{i,v}^\top$ can be computed in closed form. The parameters of the pairwise factor in viewpoint v can be computed from the 3D parameters by $\beta_{i,v} = [\mu^u_{i,v}, \mu^v_{i,v}, \sigma^u_{i,v}, \sigma^v_{i,v}, \sigma^{uv}_{i,v}]$. Analogously, the 2D displacement features are $\eta_i(\Pi_v p_0, \Pi_v p_i) = -[du, dv, du^2, dv^2, 2dudv]$.

In summary both factors define the score of a hypotheses $\mathbf{p}$ under viewpoint v for an observation I

$$\begin{aligned} \langle \boldsymbol{\theta}, \boldsymbol{\psi}(\mathbf{p}, v, I) \rangle = &\sum_{i=0}^{P} \langle \alpha_{i,v}, \phi(\Pi_v p_i, I) \rangle \\ &+ \sum_{i=1}^{P} \langle \beta_{i,v}, \eta(\Pi_v p_i, \Pi_v p_0) \rangle \end{aligned} \tag{5.10}$$

For a given 3D model y°, and its projected images $S(y^\circ)$, under viewpoints v_j the

	parts				
model	pos.	displ.	appearance	init	loss
DPM-Hinge	2D	2D	disc.	AR	hinge
DPM-VOC	2D	2D	disc.	AR	voc
DPM-VOC+VP	2D	2D	disc.	VP	vocvp
DPM-3D-Constraints	3D	2D	disc.	VP	vocvp
3D^{2}PM	3D	3D	cont.	VP	vocvp

Table 5.1: Comparison of different models in terms of part parameterization, appearance model, component initialization and training loss.

score of the 3D object hypothesis $\mathbf{p}$ is

$$\langle \boldsymbol{\theta}, \boldsymbol{\psi}(\mathbf{p}, y^\circ) \rangle = \sum_{i=0}^{P} \langle \alpha_i, \phi(p_i, y^\circ) \rangle + \sum_{i=1}^{P} \langle \beta_i, \eta(p_i, p_0) \rangle \tag{5.11}$$

Here, $\langle \alpha_i, \psi(p_i, y^\circ) \rangle$ is a 3D unary term, defined as $\langle \alpha_i, \phi(p_i, y^\circ) \rangle = \sum_{S(y^\circ)} \langle \alpha_{i,v_j}, \phi(\Pi_{v_j} p_i, I_j) \rangle$. It accumulates the 2D unary terms for every part from all projected images of the 3D model.

The 3D^{2}PM model $\boldsymbol{\theta} = [\boldsymbol{\alpha}, \boldsymbol{\beta}]$, consists of the unary parameters of the support views $[\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_K]$, as well as the parameters of the 3D displacement distribution of each part $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_P]$. Note that the 3D part displacement distributions are independent of the viewpoint components, that is every $\beta_i \in \mathbb{R}^3$. Fig. 5.3c illustrates the 3D^{2}PM model. Note that the double circle on the viewpoint variable v denotes that it is continuous. The 2D displacement parameters $\beta_{i,v}$ are obtained via projection from the 3D displacement parameters β_i, therefore they are denoted with an empty factor.

Inference. The 3D^{2}PM output are 2D or 3D object hypotheses. For an observed image I, we solve again for the MAP estimate which corresponds to the following optimization problem: $\mathbf{q}^*, v^* = \operatorname{argmax}_{v,\mathbf{p}} \langle \boldsymbol{\theta}_v, \boldsymbol{\psi}(\mathbf{p}, v, I) \rangle$ For the 3D^{2}PM-D the viewpoint variable is discrete and the inference is the same as for DPM-3D-Constraints.

The MAP inference problem for 3D^{2}PM-C is a continuous problem. In practice we choose, at test time, an arbitrarily fine viewpoint binning. After this discretization, we proceed with the same inference procedure as for 3D^{2}PM-D. Note that this is different from choosing a viewpoint discretization at training time. This model allows to estimate the viewpoint up to an arbitrary precision, only chosen at test time.

For a 3D example y°, the inference problem is $\mathbf{p}^* = \operatorname{argmax}_{\mathbf{p}} \langle \boldsymbol{\theta}, \boldsymbol{\psi}(\mathbf{p}, y^\circ) \rangle$, that is a consistent output is required for all images of the same instance. First, all unary terms are computed by collecting evidence from all available image projections of y°, then, the 3D distance transform can be used to solve for optimal part placements in 3D.

Learning. Real images and additionally 3D models are used for training. We assume that the training data comes with angular accurate viewpoint annotations

$y^v \in [0^\circ, 360^\circ)$. The 3D^{2}PM and all of its variants are learned using the same regularized risk objective as the DPM-VOC+VP, described in Eq. (5.5). Again, the loss measures detection and viewpoint estimation, with the difference that the loss reflects the continuous viewpoint estimate $\Delta_{VP}(y, \bar{y}) = \angle(y^v, \bar{y}^v)/180$. Note that the learning algorithm does not need to change, we use Alg. 1 with the only modification in line 17. To obtain the gradients w.r.t. the 3D parameters β, we simply take the gradients via the projected parameters using the chain rule. Tab. 5.1 summarizes the qualitative differences among the different models.

Rendering CAD models. We use the non-photorealistic wireframe-like renderings of (Stark *et al.*, 2012), using a perspective projection. Depending on the dataset, we render all the CAD models in $\{8, 12, 16, 18, 24, 36\}$ equally spaced viewpoints, independently from the viewpoint statistics of the given dataset.

5.3 EXPERIMENTS

In this section, we thoroughly evaluate our models on various datasets measuring their performance in terms of 2D BB localization, viewpoint estimation, and, in the case of DPM-3D-Constraints and 3D^{2}PM, their ability to predict part that correspond across viewpoints. To that end, we follow the ordering of Sect. 5.2, and successively add 3D information to the models under consideration.

We start by analyzing the performance of our structured output learning framework in comparison to the standard DPM formulation (Felzenszwalb *et al.*, 2010), highlighting its ability to provide both better BB localization (DPM-VOC) and simultaneous viewpoint estimation (DPM-VOC+VP) (Sect. 5.3.2). Second, we examine the impact of parameterizing object parts in 3D object coordinates rather than in the 2D image plane (DPM-3D-Constraints and 3D^{2}PM), again for the task of 2D BB localization and viewpoint estimation (Sect. 5.3.3), demonstrating superior performance in comparison to both previous work in 3D object class modeling and the standard DPM (Felzenszwalb *et al.*, 2010). Third, we leverage the ability of our 3D^{2}PM model to predict viewpoints of arbitrary granularity for fine-grained viewpoint estimation (Sect. 5.3.4), again outperforming prior work. And fourth, as shown in chapters 3 and 4,we apply DPM-3D-Constraints and 3D^{2}PM to the task of ultra-wide baseline matching, quantifying their ability to localize corresponding parts in multiple views of the same object.

All experiments are conducted on publicly available standard benchmarks for the respective task (Sect. 5.3.1) and include extensive comparisons to previous work.

5.3.1 Data sets

We commence with a brief overview of the five diverse datasets used in the experiments.

Pascal VOC 2007. The detection benchmark of the Pascal VOC suite (Everingham *et al.*, 2010) provides a challenging test bed for 2D bounding box localization

of 20 object classes. It is considered challenging due to strong variations in object appearance, background clutter, and partial occlusion. The 2007 version (Everingham *et al.*, 2007) has emerged as the standard benchmark for object detection approaches.

Pascal3D+. Recently, 12 object classes of Pascal VOC 2012 have been enriched with additional viewpoint annotations (Xiang *et al.*, 2014a) by fitting 3D CAD models to images in a semi-automatic procedure. The performance is measured in terms of simultaneous 2D BB localization and viewpoint estimation. A candidate detection can only qualify as a true positive if it satisfies both the VOC intersection-over-union criterion (Everingham *et al.*, 2010) and provides correct viewpoint class estimate. We refer to the joint metric as average viewpoint precision (AVP).

3D Object Classes. Introduced in 2007, the 3D Object Classes dataset (Savarese and Fei-Fei, 2007) still constitutes the de-facto standard dataset for multi-view recognition (i.e., 2D BB localization and viewpoint estimation). It provides images of nine object classes taken under controlled conditions w.r.t. viewpoint (three discrete different camera distances, three elevations, and eight azimuth angles) but exhibiting considerable background clutter and challenging lighting variations. Viewpoint estimation on this dataset is typically phrased as an 8-class classification problem (one class per azimuth angle).

EPFL Multi-view cars. This dataset (Ozuysal *et al.*, 2009) has been recorded in the course of a car exhibition, where cars are presented to the audience on rotating platforms. While it features only a single object per image, lighting conditions are challenging (bright lights lead to specularities and saturation effects). Viewpoint annotations are almost angle-accurate (derived from platform rotation speed) and provide for a challenging fine-grained viewpoint estimation benchmark.

KITTI. The KITTI dataset (Geiger *et al.*, 2012) has been recorded from a moving vehicle driving through the city of Karlsruhe. It comes with manual BB and viewpoint annotations derived from 3D Lidar scans. It is challenging due to significant amounts of occlusion.

5.3.2 Structured output learning

We first compare the performance of our structured output learning framework (DPM-VOC, DPM-VOC+VP, Sect. 5.2.3) to the standard DPM. We evaluate on the following three data sets: Pascal VOC 2007, 3D Object Classes, and Pascal3D+. In all experiments, we use images from the respective data sets for training, following the protocols established as part of the data sets.

2D Bounding box localization. Tab. 3.1 gives results for 2D BB localization on the Pascal VOC 2007 dataset, according to the Pascal criterion, reporting per-class average precision (AP). It compares our DPM-VOC (row 2) to the DPM-Hinge (Felzenszwalb *et al.*, 2009) (row 1) and to the multi-kernel learning approach of Vedaldi *et al.* (2009) (row 3), both of which are considered to be among the state-of-the-art on this data set. We first observe that DPM-VOC outperforms DPM-Hinge on 18 of 20 classes, and Vedaldi *et al.* (2009) on 8 classes. While the relative performance

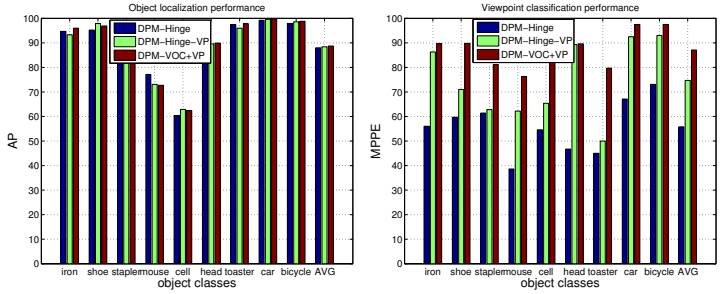


Figure 5.4: 2D bounding box localization (left) and viewpoint estimation (right) results on nine 3D Object classes (Savarese and Fei-Fei, 2007).

difference of 1.1% on average (31.4% AP vs. 30.3% AP) to DPM-Hinge is moderate in terms of numbers, it is consistent and speaks in favor of the structured loss over the standard hinge loss. In comparison to Vedaldi *et al.* (32.1% AP), DPM-VOC loses only 0.7% while the DPM-Hinge has 1.8% lower AP. We note that Vedaldi *et al.* exploits a variety of different features for performance, while the DPM-VOC and DPM-Hinge use HOG features only.

Fig. 5.4 (left) gives the corresponding results for 9 3D Object Classes, comparing DPM-Hinge (col. 1), DPM-Hinge-VP (col. 2), and DPM-VOC+VP (col. 3), where we initialize and fix each component of the DPM-Hinge with training data from just a single viewpoint, identical to DPM-VOC+VP. We observe a clear performance ordering, improving from DPM-Hinge over DPM-Hinge-VP to DPM-VOC+VP, which wins for 5 of 9 classes. While the average improvement is again moderate (performance increases from 88.0% over 88.4% to 88.7% AP), it confirms the benefit of the structured output objective, compared to the classification one.

Viewpoint estimation. Figure 5.4 (right) gives results for viewpoint estimation, phrased as a classification problem, distinguishing among eight distinct azimuth angle classes. In line with previous work (Savarese and Fei-Fei, 2007; Lopez-Sastre *et al.*, 2011), we report the mean precision in pose estimation (MPPE) on true positive detections according to the Pascal criterion (equivalent to the average over the diagonal of the confusion matrix). While there is an explicit association between mixture components and viewpoints for DPM-Hinge-VP and DPM-VOC+VP, we let the DPM-Hinge predict the most likely viewpoint by collecting votes from training example annotations for each component.

Clearly, the explicit association between viewpoints and mixture components already helps significantly (74.7% DPM-Hinge-VP vs. 55.8% DPM-Hinge), but we achieve a further boost by 12.4% in performance by applying a structured rather than hinge-loss (87.1% DPM-VOC+VP vs 74.7% DPM-Hinge-VP). A nice side effect is that training becomes considerably faster when fixing the mixture component assignments.

Simultaneous BB localization and VP estimation. So far, we have evaluated viewpoint estimation under rather special conditions. We have considered with 3D Object Classes a dataset for which 2D BB localization performance has essentially

AP/AVP	**aeroplane**	**bicycle**	**boat**	**bus**	**car**	**chair**
DPM-Hinge-4V	35.3 / 20.3	**47.8** / 26.2	**3.6** / **2.0**	**52.3** / **49.8**	35.1 / 24.8	**13.9** / **6.9**
DPM-Hinge-8V	35.6 / 3.9	45.7 / 6.4	**6.3** / **1.2**	48.1 / **44.9**	38.1 / 17.0	**14.2** / 3.6
DPM-Hinge-16V	33.4 / 1.0	43.1 / 1.0	3.9 / 0.3	44.9 / 26.7	36.7 / 6.9	**15.3** / 1.5
DPM-Hinge-24V	28.7 / 0.3	41.1 / 0.4	3.7 / 0.3	38.8 / 4.9	35.6 / 2.6	**13.0** / 0.8
VDPM - 4V	40.0 / 34.6	45.2 / 41.7	3.0 / 1.5	49.3 / 26.1	37.2 / 20.2	11.1 / 6.8
VDPM - 8V	39.8 / 23.4	47.3 / 36.5	5.8 / 1.0	50.2 / 35.5	37.3 / 23.5	11.4 / 5.8
VDPM - 16V	**43.6** / 15.4	**46.5** / 18.4	**6.2** / 0.5	54.6 / 46.9	36.6 / 18.1	12.8 / 6.0
VDPM - 24V	**42.2** / 8.0	44.4 / 14.3	**6.0** / 0.3	53.7 / 39.2	36.3 / 13.7	12.6 / 4.4
DPM-VOC+VP - 4V	**43.8** / **39.4**	47.0 / **43.9**	0.5 / 0.3	51.7 / 49.1	**46.3** / **37.6**	9.2 / 6.1
DPM-VOC+VP - 8V	**42.0** / **29.7**	**49.8** / **42.6**	0.9 / 0.4	**52.0** / 39.5	**47.9** / **36.8**	11.3 / **9.4**
DPM-VOC+VP - 16V	39.3 / **17.0**	46.3 / **24.7**	2.6 / **1.0**	**55.3** / **49.0**	**46.0** / **30.1**	10.4 / **6.6**
DPM-VOC+VP - 24V	37.7 / **10.6**	**45.9** / **16.7**	5.6 / **2.2**	**55.2** / **43.5**	**42.9** / **25.4**	9.1 / **4.4**

	diningtable	**motorbike**	**sofa**	**train**	**tvmonitor**	**Avg.**
DPM-Hinge-4V	**9.9** / **9.5**	**39.8** / 23.0	10.7 / 10.3	**26.7** / **23.9**	34.9 / **34.8**	**28.2** / 21.1
DPM-Hinge-8V	9.2 / 4.0	34.3 / 5.9	5.6 / 4.4	24.2 / **20.8**	33.3 / **32.7**	26.8 / 13.2
DPM-Hinge-16V	5.8 / 1.8	32.7 / 1.0	11.0 / 6.1	21.8 / 16.1	30.5 / 20.0	25.4 / 7.5
DPM-Hinge-24V	8.2 / 2.0	30.1 / 1.0	10.1 / 4.9	21.3 / 6.2	28.1 / 9.0	23.5 / 3.0
VDPM - 4V	7.2 / 3.1	33.0 / 30.4	6.8 / 5.1	26.4 / 10.7	**35.9** / 34.7	26.8 / 19.5
VDPM - 8V	**10.2** / **3.6**	36.6 / 25.1	**16.0** / **12.5**	**28.7** / 10.9	**36.3** / 27.4	**29.9** / 18.7
VDPM - 16V	**7.6** / 2.2	38.5 / 16.1	**16.2** / **10.0**	**31.5** / **22.1**	**35.6** / 16.3	**30.0** / 15.6
VDPM - 24V	**11.1** / **3.6**	35.5 / 10.1	**17.0** / **8.2**	**32.6** / 20.0	**33.6** / 11.2	**29.5** / 12.1
DPM-VOC+VP - 4V	5.7 / 3.0	34.7 / **32.2**	**13.3** / **11.8**	17.4 / 12.5	33.4 / 33.2	27.5 / **24.5**
DPM-VOC+VP - 8V	5.3 / 2.6	**39.8** / **32.9**	13.5 / 11.0	21.4 / 10.3	33.1 / 28.6	28.8 / **22.2**
DPM-VOC+VP - 16V	7.5 / **3.0**	**39.5** / **17.2**	12.7 / 7.7	28.5 / 20.4	30.7 / **20.2**	29.0 / **17.9**
DPM-VOC+VP - 24V	7.6 / 2.3	**35.7** / **11.3**	11.5 / 4.9	31.1 / **22.4**	27.6 / **14.4**	28.2 / **14.4**

Table 5.2: The results of DPM-Hinge, VDPM and DPM-VOC+VP are shown. The first number indicates the Average Precision (AP) for detection and the second number shows the AVP for joint object detection and pose estimation.

saturated beyond 95% AP for many classes. Then, we have evaluated viewpoint estimation entirely separately from 2D BB localization, on successful detections. While this is in line with standard evaluation procedures and prior work, it seems artificial for higher level applications, such as scene-understanding, or object tracking which require to solve both tasks simultaneously.

We hence turn to the recently proposed Pascal3D+ dataset that is both highly challenging in terms of 2D BB localization and comes with viewpoint annotations that allow to evaluate AVP (Sect. 5.3.1) in four different granularities (4, 8, 16, and 24 viewpoint classes). As baselines we again use DPM-Hinge, as well as the VDPM introduced in Xiang *et al.* (2014a). The VDPM is a viewpoint initialized DPM-Hinge (similarly to DPM-Hinge-VP), except that Xiang *et al.* (2014a) flips the viewpoint components, resulting in twice as many components compared to DPM-VOC+VP.

Table 5.2 provides the corresponding AVP results and also gives separate 2D BB localization AP results as a reference. In terms of AVP, DPM-VOC+VP (24.5%, 22.2%, 17.9%, and 14.4% for the 4 different viewpoint granularities) outperforms both the VDPM (19.5%, 18.7%, 15.6%, 12.1%) and the DPM-Hinge (21.1%, 13.2%, 7.5%, 3.0%) by large margins, for all viewpoint granularities (it improves over the VDPM by 5.0%, 3.5%, 2.3%, and 2.3% respectively, and over the DPM-Hinge by 3.4%, 9.0%, 10.4% and 11.4%). Interestingly, DPM-VOC+VP can better deal with opposing object viewpoints than VDPM, since it explicitly incorporates the viewpoint loss.

In terms of pure 2D BB localization, our DPM-VOC+VP with 27.5%, 28.8%, 29.0%, 28.2% outperforms the DPM-Hinge (28.2%, 26.8%, 25.4%, 23.5%) on 3 of 4 viewpoint granularities. Compared to VDPM (26.8%, 29.9%, 30.0%, 29.5%), DPM-VOC+VP is slightly worse (0.7% on average), which can be attributed to the fact that VDPM flips the viewpoint components, thus effectively having two components per viewpoint.

5.3.3 3D Object class representations

In the previous section, we confirmed improvements from the structured output learning framework for 2D BB localization and viewpoint estimation over the standard DPM. Here, we analyze the impact of adding 3D information, by first introducing a 3D part parameterization (DPM-3D-Constraints) and then adding 3D part displacement and continuous appearance models ($3D^2PM$). Experiments are conducted on 3D Object Classes, KITTI and Pascal VOC 2007.

In addition, we examine the effect of adding synthetic data in the form of rendered 3D CAD models (Sect. 5.2.4) to the respective training sets of real-world images, resulting in two different training data settings: (i) real data only, and (ii) mixed data (real and synthetic). Please note that both DPM-3D-Constraints and $3D^2PM$ always employ 3D CAD models for establishing a 3D coordinate system, irrespective of whether synthetic images are used for training appearance models.

In contrast to chapters 3 and 4 we extend the evaluation to the KITTI and the 3D object classes (report performance on all categories). While the models on 3D object classes in Chapter 3 modeled only the azimuth variable, here we also represent the elevation angle, resulting in more detailed and more powerful representations.

	2D Models			
AP/MPPE	DPM-VOC+VP	Lopez-Sastre *et al.*	Bao *et al.*	Payet and Todorovic
car	**99.9/ 97.9**	96.0/ 87.9	98.0/ 95.3	-/ 86.1
bicycle	**98.8/ 97.5**	91.0/ 89.9	93.1/ 92.3	-/ 80.8
iron	**98.1/ 94.2**	53.0/ 90.8	82.5/ 89.8	-/ -
shoe	**98.8/ 97.6**	78.0/ 89.3	85.5/ 88.0	-/ -
stapler	**89.8/ 92.6**	32.0/ 79.3	70.2/ 73.9	-/ -
mouse	**77.4** / 82.0	41.0/ 66.4	54.5/ 72.0	-/ -
cell.	71.4/ 90.7	43.0/ 75.4	**81.0**/ 86.0	-/ -
head	**90.9** / **90.7**	76.0/ 77.4	- / -	-/ -
toaster	97.0/ **81.6**	54.0/ 56.9	**98.2**/ 70.3	-/ -
avg	**91.3/ 91.5**	62.7/ 79.2	83.0/ 83.5	-/ -

	3D Models						
AP/MPPE	$3D^2PM$	DPM-3D-Constr.	Xiang and Savarese	Yoruk and Vidal	Liebelt and Schmid	Zia *et al.*	Glasner *et al.*
car	99.8/ 97.5	99.4/ 97.5	98.3/ 93.1	93.3/ 73.0	76.7/ 70.0	90.4/ 84.0	99.2/ 85.3
bicycle	96.6/ 96.4	95.8/ 96.1	93.8/ 90.1	-/ -	69.8/ 75.5	-/ -	-/ -
iron	97.2/ 93.1	97.7/ 93.6	82.2/ 86.0	-/ -	-/ -	-/ -	-/ -
shoe	98.3/ 95.8	97.9/ 96.1	84.1/ 86.6	-/ -	-/ -	-/ -	-/ -
stapler	88.4/ 86.9	86.4/ 89.1	70.5/ 73.2	-/ -	-/ -	-/ -	-/ -
mouse	74.9/ 83.5	77.0/ **88.3**	52.2/ 69.8	-/ -	-/ -	-/ -	-/ -
cell.	70.1/ 91.2	67.4/ **92.7**	80.2/ 86.3	-/ -	-/ -	-/ -	-/ -
head	92.5/ 88.7	88.5/ 88.7	- / -	-/ -	-/ -	-/ -	-/ -
toaster	95.4/ 71.9	96.4/ 78.1	97.5/ 65.4	-/ -	-/ -	-/ -	-/ -
avg	90.4/ 89.4	89.6/ 91.1	82.3/ 81.3	-/ -	-/ -	-/ -	-/ -

Table 5.3: Comparison to state-of-the-art in 2D BB localization and viewpoint estimation on 3D Object classes (Savarese and Fei-Fei, 2007).

AP/MPPE	real			mixed		
	DPM-VOC+VP	DPM-3D-Constr.	3D^{2}PM	DPM-VOC+VP	DPM-3D-Constr.	3D^{2}PM
car	**99.9** / **97.9**	99.4 / 97.5	99.8 / 97.5	**99.9** / **97.9**	99.7 / 96.3	99.6 / 97.1
bicycle	**98.8** / **97.5**	95.8 / 96.1	96.6 / 96.4	**98.8** / 97.5	95.0 / 96.4	96.2 / **97.5**
iron	**98.0** / **94.9**	97.7 / 93.6	97.2 / 93.1	**98.1** / 94.2	97.7 /93.6	97.1 / **95.6**
shoe	**99.4** / **97.0**	97.9 / 96.1	98.3 / 95.8	**98.8** / **97.6**	97.0 / 95.8	97.2 / 94.3
stapler	**90.3** / **89.5**	86.4 / 89.1	88.4 / 86.9	**89.8** / **92.6**	85.6 / 85.6	79.9 / 84.3
mouse	75.3 / 84.2	**77.0** / **88.3**	74.9 / 83.5	77.4 / 82.0	**79.0** / **86.5**	69.9 / 79.2
cellphone	**73.9** / **93.5**	67.4 / 92.7	70.1 / 91.2	**71.4** / 90.7	65.7 / **92.4**	69.0 / 91.7
head	89.9 / **89.6**	88.5 / 88.7	**92.5** / 88.7	90.9 / 90.7	88.7 / 86.4	**92.1** / **93.0**
toaster	96.3 / **79.4**	**96.4** / 78.1	95.4 / 71.9	**97.0** / **81.6**	95.1 / 74.9	96.9 / 76.7
avg	**91.3** / 91.5	89.6 / 91.1	90.4 / 89.4	**91.3** / **91.6**	89.3 / 89.8	88.7 / 89.9

Table 5.4: Viewpoint estimation and object localization results using real and mixed training data on 3D Object Classes (Savarese and Fei-Fei, 2007), comparing our different models.

3D Object Classes. Tab. 5.4 compares the performance of our DPM-VOC+VP, DPM-3D-Constraints and 3D^{2}PM models, in the real and mixed data settings on the 3D object classes dataset. From the average results across all the classes, we observe that in the real data setting, DPM-VOC+VP with 91.3% AP and 91.5% MPPE, is slightly better than 3D^{2}PM (90.4% AP, 89.4% MPPE), which in turn has comparable performance to the DPM-3D-Constraints (89.6% AP, 91.1% MPPE). In the mixed data setup, the results have the same tendencies. Again, DPM-VOC+VP with 91.3% AP and 91.6% MPPE is slightly better than DPM-3D-Constraints (89.3% AP, 89.8% MPPE) which is comparable to the 3D^{2}PM (88.7% AP, 89.9% MPPE). The slightly better performance by the DPM-VOC+VP does not come at a surprise, since it directly optimizes for the task at hand, namely 2D BB localization and viewpoint estimation, whereas DPM-3D-Constraints and 3D^{2}PM successively introduce 3D geometric constraints into the model during learning, which come at the cost of slightly worse performance. Comparing the two different data settings, although one would expect that more data always helps, the mix data setup does not reflect that intuition due to the unrealistic appearance of the synthetically generated data.

Table 5.3 compares the DPM-VOC+VP, DPM-3D-Constraints, and 3D^{2}PM with state-of-the-art results on 3D Object Classes (Savarese and Fei-Fei, 2007), distinguishing 2D and 3D object class representations. We make the following observations. First, DPM-VOC+VP (91.3% AP, 91.6% MPPE) outperforms all other methods on average (last row) as well as on 6 of 9 classes. It outperforms the next best prior result of 82.3% AP and 81.3% MPPE obtained by the Aspect Layout Model (ALM) (Xiang and Savarese, 2012) by 9.0% and 10.3% respectively, despite the ALM making use of additional human annotation in the form of aspect layout parts. Second, the top performance of DPM-VOC+VP is almost matched by both of our 3D object class representations, DPM-3D-Constraints (89.6% AP, 91.1% MPPE) and 3D^{2}PM (90.4% AP, 89.4% MPPE). This is remarkable since the 3D representations put additional (3D) constraints on the learned model, while DPM-VOC+VP is only bound by the combined localization and viewpoint loss, directly optimizing for the task at hand

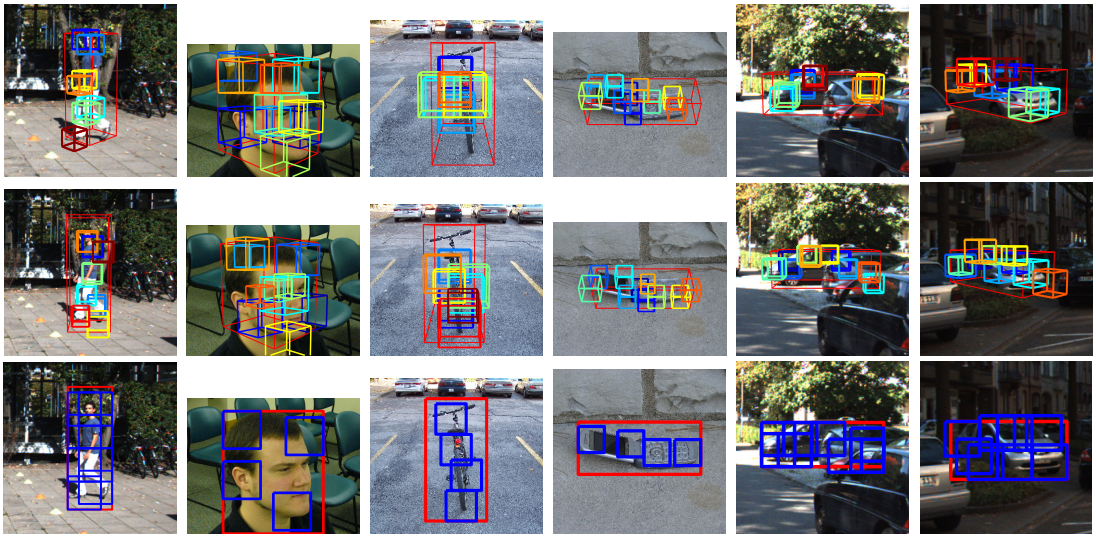


Figure 5.5: Qualitative results on KITTI and 3D object classes. Corresponding part detections (for a given class) are color coded. 3D^{2}PM (first row), DPM-3D-Constraints (second row) and DPM-VOC+VP (third row).

without any additional constraints. And third, we see that our models also compare favorably to prior work that has specialized on certain object classes, such as cars. Specifically, 3D^{2}PM outperforms the voting-based approach of Glasner *et al.* (2011) (99.2% AP, 85.3% MPPE) which relies on 3D reconstructions of the object class of interest as training data.

KITTI. Table 5.6 provides 2D BB localization and viewpoint estimation results on the challenging KITTI dataset for our models, trained from either purely real or mixed training data. We split the dataset into three equal sets, used for training, validation and testing. Starting with the real data setting, we observe that all our models consistently outperform the DPM-Hinge: the improvements in average AP range from 0.6% (3D^{2}PM) over 2.2% (DPM-3D-Constraints) to 5.8% (DPM-VOC+VP) and in MPPE from as much as 22.3% (3D^{2}PM) over 26.9% (DPM-3D-Constraints) to 29.3% (DPM-VOC+VP). Comparing our different models, the DPM-VOC+VP performs best (47.7% AP, 54.3% MPPE), followed by DPM-3D-Constraints (44.1% AP, 51.9% MPPE) and 3D^{2}PM (42.5% AP and 47.3% MPPE) – it seems that the added expressiveness of our 3D models DPM-3D-Constraints and 3D^{2}PM comes at a (moderate) cost w.r.t. performance, which we attribute to occlusion. 3D^{2}PM performs worse on medium to highly occluded objects, compared to DPM-VOC+VP. On the 0-20% occlusion level, 3D^{2}PM (79.2%) achieves 0.3% better performance than DPM-VOC+VP (78.9%), but on the rest of the occlusion levels it is consistently worse (e.g. on 60-80% DPM-VOC+VP is better by 3.0%).

Adding synthetic training images improves the performance of our models mostly for viewpoint estimation: DPM-3D-Constraints improves by 2.4%, from 51.9% to 54.3% MPPE, and 3D^{2}PM from 47.3% to 47.7% MPPE. For 2D BB localization, only 3D^{2}PM improves by 1.5% from 42.5% to 44.0% AP, while the other models lose

AP/AOS	DPM-VOC+VP	DPM-3D-Constr.	3D^{2}PM	DPM-Hinge
car	48.8/46.5	42.2/40.1	45.6/42.9	41.0/-
ped.	40.8/35.7	36.6/29.6	37.4/30.7	34.8/-
cycl.	28.2/21.6	27.5/21.1	27.1/20.9	27.3/-
avg	39.3/34.6	35.4/30.3	36.7/31.5	34.4/

Table 5.5: 2D BB localization and viewpoint estimation on KITTI testing (Geiger *et al.*, 2012).

AP/MPPE	real			mixed			baseline
	DPM-VOC+VP	DPM-3D-Constr.	3D^{2}PM	DPM-VOC+VP	DPM-3D-Constr.	3D^{2}PM	DPM-Hinge
car	**63.0** / **73.6**	61.6 / 70.7	60.3 / 63.2	61.4 / 70.8	60.8 / 71.3	61.3 / 65.6	60.5 / 46.1
pedestrian	43.7 / **46.6**	38.0 / 31.9	36.1 / 40.0	**43.9** / 45.4	35.9 / 45.6	38.9 / 41.3	36.2 / 22.9
cyclist	**36.5** / 42.6	32.7 / **53.0**	31.1 / 38.8	36.3 / 46.8	30.4 / 45.9	31.8 / 36.1	28.9 / 6.0
AVG	**47.7** / 54.3	44.1 / 51.9	42.5 / 47.3	47.2 / 54.3	42.4 / **54.3**	44.0 / 47.7	41.9 / 25.0

Table 5.6: 2D BB localization and viewpoint estimation on KITTI (Geiger *et al.*, 2012).

performance (DPM-VOC+VP loses 0.5% AP, DPM-3D-Constraints loses 1.7%). We attribute this mixed behavior to the fact that synthetic training images come with perfect, angular accurate viewpoint annotations (improving viewpoint estimation), but often deviate from real-world training images in terms of appearance, at least for the chosen type of edge-based rendering – we leave improving the rendering quality for future work. Fig. 5.5 shows qualitative results on KITTI and 3D object classes.

Table 5.5 shows the results in terms of AP and AOS (average orientation similarity) (Geiger *et al.*, 2012), now on the KITTI testing set (Geiger *et al.*, 2012). DPM-VOC+VP (39.3%), DPM-3D-Constraints (35.4%) and 3D^{2}PM (36.7%) outperform the DPM-Hinge (34.4%) across all the classes.

5.3.4 3D Deformations and continuous appearance

While accurate 2D BB localization and viewpoint classification into coarse classes can be achieved with a purely view-based 2D (DPM-VOC+VP, Sect. 5.3.2) or 3D (DPM-3D-Constraints, Sect. 5.3.3) object class representation, estimating viewpoint on a finer level of granularity demands a proper 3D object class model with 3D deformations and continuous appearance, such as 3D^{2}PM. In this section, we hence highlight the ability of our 3D^{2}PM to predict viewpoint up to arbitrary granularity. To that end, we use the EPFL Multi-view cars dataset (Sect.5.3.1), due to its angle-accurate viewpoint annotations and uniform sampling of the viewing circle.

In comparison to Chapter 4, we change the model learning to include specialized regularization multipliers and explicit bounds for the 3D pairwise parameters.

Arbitrarily fine viewpoint estimation. In order to assess the ability of our 3D^{2}PM models to generate viewpoint estimates of arbitrarily fine granularity, we train 3D^{2}PM-C with a varying number of $k \in \{8, 12, 16, 18, 36\}$ support views, interpolating to a varying number of predicted views of increasing resolution $d \in$

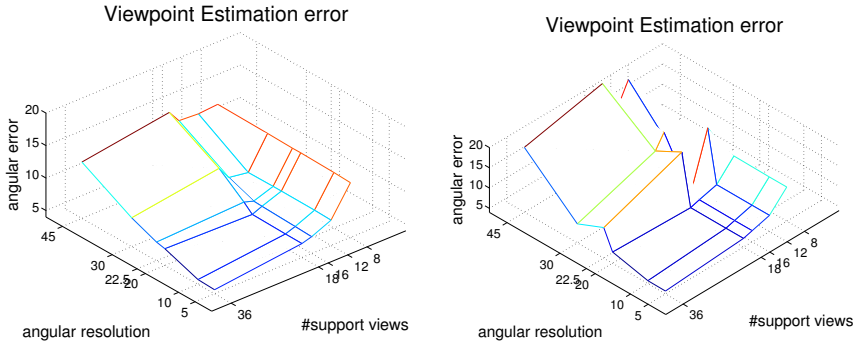


Figure 5.6: Fine viewpoint estimation performance (in MAE) using linear (left) and exponential interpolation (right).

MAE	Glasner *et al.*	3D^{2}PM-D	3D^{2}PM-C-Lin	3D^{2}PM-C-Exp
8 bins	24.80	12.89	**12.43**	12.63
12 bins	-	7.99	**7.89**	7.99
16 bins	-	7.00	**6.59**	6.77
18 bins	-	6.29	**6.15**	6.15
36 bins	-	4.74	**4.62**	4.70

Table 5.7: Fine viewpoint estimation on EPFL (Ozuysal *et al.*, 2009).

$\{45°, 30°, 22.5°, 20°, 10°, 8°, 5°\}$. Fig. 5.6 plots the corresponding results for 3D^{2}PM-C-Lin (left) and 3D^{2}PM-C-Exp (right) as surfaces of MAE over k and d.

For both models, we observe that both, increasing k for fixed d and decreasing d for fixed k, in fact results in lower angular error in most cases, highlighting the benefit of the 3D continuous representation. The respective minima are attained at $k = 36$, $d = 5°$ (4.62° MAE for 3D^{2}PM-C-Lin and 4.70° for 3D^{2}PM-C-Exp), approaching the dataset viewpoint label noise.

Comparison to state-of-the-art. Table 5.7 reports MAE for our 3D^{2}PM models at 5° resolution comparing to state-of-the-art. The 3D^{2}PM-D, 3D^{2}PM-C-Lin, and 3D^{2}PM-C-Exp models with $k = 8$ achieve 12.89°, 12.43° and 12.63° MAE outperforming by almost 12° the best published result of 24.8° of Glasner *et al.* (2011).

Table 5.8 gives a comparison to prior results that had been measured in terms of 2D BB localization (AP) and viewpoint estimation performance (MPPE) rather

AP/MPPE	3D^{2}PM-D	3D^{2}PM-C-Lin	3D^{2}PM-C-Exp	Xiang and Savarese	Ozuysal *et al.*	Lopez-Sastre *et al.*
8 bins	**99.8**/77.6	99.7/**80.6**	98.8/79.4	-/-	-/-	91.0/73.7
12 bins	98.9/79.0	**99.6**/**83.1**	99.5/81.1	-/-	-/-	- / -
16 bins	99.8/70.8	**99.8**/73.5	99.6/**74.0**	98.1/56.6	85.0/41.6	97.0/66.0
18 bins	99.8/72.1	99.8/**75.0**	**99.9**/73.8	-/-	-/-	-/-
36 bins	99.9/52.7	**99.9**/**55.9**	99.7/54.5	-/-	-/-	-/-

Table 5.8: 2D BB localization (AP) and viewpoint estimation (MPPE (Lopez-Sastre *et al.*, 2011)) on EPFL (Ozuysal *et al.*, 2009).

than MAE (note that MPPE is measured according to the respective number of support views and is not comparable across table rows). We observe that our models outperform prior results in AP and MPPE by significant margins. 3D^{2}PM-C-Lin (99.7% AP, 80.6% MPPE) performs best on average, outperforming Lopez-Sastre *et al.* (2011) (91.0% AP, 73.7% MPPE) by 8.7% and 6.9% for 8 support views, and by 1.8% and 7.5% for 16 views, respectively. Interpolation (3D^{2}PM-C-Lin and 3D^{2}PM-C-Exp) consistently improves performance by around $2-3\%$ over 3D^{2}PM-D in terms of MPPE, and 3D^{2}PM-C-Lin is around $1-2\%$ better than 3D^{2}PM-C-Exp on average.

5.4 CONCLUSION

In this chapter, we presented our multi-view and 3D object representations in a unified framework and provided additional strong experimental evaluation across several datasets and object categories, confirming the benefits of the 3D object representation. In particular, we extended the DPM (Felzenszwalb *et al.*, 2010) to include viewpoint and 3D geometry information, thus bringing the world of 2D object detectors and 3D object representations closer. By adding 3D geometry information on three different levels (viewpoints, part parameterization and viewpoint continuous appearance), in this work we have provided a palette of object detectors, which gradually and successfully introduce object geometry into the DPM. The 3D^{2}PM extends the DPM to a full 3D object model. It leverages 3D information from CAD data, performing viewpoint estimation at arbitrarily fine granularity.

In an extensive experimental study on several datasets with varying level of difficulty, and on several different classes we have shown that the presented models achieve state-of-the-art performance in terms of viewpoint estimation and ultra-wide baseline part matching (Chapter 4), while maintaining competitive object localization performance, confirming the ability to deliver expressive object hypotheses.

3 D OBJECT CLASS DETECTION IN THE WILD 6

Contents

Estimating the 3D shape, 3D pose and 3D position of objects has been a long standing goal in computer vision. While the previous chapters presented methods with a coarse 3D shape representation (3D star-CRF), in this chapter we introduce a much more detailed 3D shape representation. Namely, we draw from recent advances in object detection and 2D-3D object lifting in order to design method particularly tailored towards 3D object class detection. The presented 3D object class detection method assumes single image as input and consists of several stages gradually enriching the object detection output with object viewpoint, keypoints and 3D shape estimates. The final result is an aligned computer aided design (CAD) model to objects in images, resulting in a very rich and detailed 3D shape representation. Relying on convolutional neural networks (CNNs), the presented method can reliably detect objects in 3D from various object categories in challenging real world scenarios.

6.1 introduction

Deliniating the content of a visual scene, object by object in 3D, from just a single image has been a long standing goal of computer vision since its early days (Marr and Nishihara, 1978; Brooks, 1981; Pentland, 1986; Lowe, 1987). It has been argued that higher-level tasks, such as scene understanding or object tracking, can benefit from detailed, 3D object hypotheses (Ess *et al.*, 2009; Wojek *et al.*, 2010; Geiger *et al.*, 2014) that allow to explicitly reason about occlusion (Zia *et al.*, 2013b; Bo Li and Zhu,

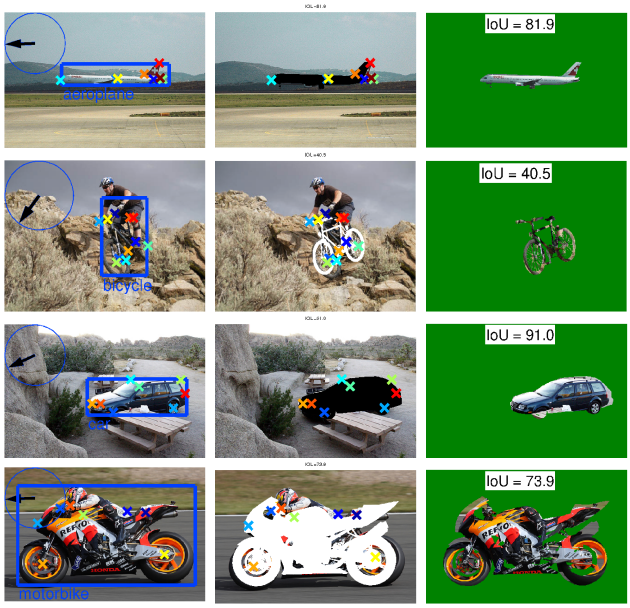


Figure 6.1: Output of our 3D object class detection method. (Left) BB, keypoints and viewpoint estimates, (center) aligned 3D CAD prototype, (right) segmentation mask.

2014) or establish correspondences across multiple frames (Xiang *et al.*, 2014b). As a consequence, there has been an increasing interest in designing object class detectors that predict more information than just 2D bounding boxes, ranging from additional viewpoint estimates (Stark *et al.*, 2010; Gu and Ren, 2010; Lopez-Sastre *et al.*, 2011; Xiang *et al.*, 2014a) over 3D parts that correspond across viewpoints (Thomas *et al.*, 2006) to the precise 3D shape of the object instance observed in a test image (Zia *et al.*, 2013a; Yoruk and Vidal, 2013; M.Hejrati and D.Ramanan, 2012).

So far, these efforts have lead to two main results. First, it has been shown that simultaneous 2D bounding box localization and viewpoint estimation, often in the form of classification into angular bins, are feasible for rigid object classes (Thomas *et al.*, 2006; Savarese and Fei-Fei, 2007; Liebelt *et al.*, 2008; Ozuysal *et al.*, 2009; Arie-Nachimson and Basri, 2009; Liebelt and Schmid, 2010). These *multi-view object class detectors* typically use view-based (Lopez-Sastre *et al.*, 2011) or coarse 3D geometric (Xiang and Savarese, 2012; Fidler *et al.*, 2012; M.Hejrati and D.Ramanan, 2012; Hejrati and Ramanan, 2014) object class representations that are designed to generalize across variations in object shape and appearance. While these representations have shown remarkable performance through the use of joint training with structured losses (Chapter 3), they are limited with respect to the provided geometric detail.

Second, and more recently, it has been shown that highly detailed 3D shape hypotheses can be obtained by *aligning 3D CAD model instances* to an image (Zia *et al.*, 2013a; Lim *et al.*, 2013; Aubry *et al.*, 2014; Lim *et al.*, 2014). These approaches are based on a large database of 3D CAD models that ideally spans the entire space of object instances expected at recognition time. Unfortunately, the added detail comes at a cost: first, these approaches are targeted only towards specific object classes

like cars and bicycles (Zia *et al.*, 2013a), chairs (Aubry *et al.*, 2014), or pieces of IKEA furniture (Lim *et al.*, 2013, 2014), limiting their generality. Second, they are typically evaluated on datasets with limited clutter and occlusion (Zia *et al.*, 2013a), such as 3D Object Classes (Savarese and Fei-Fei, 2007), EPFL Multi-View Cars (Ozuysal *et al.*, 2009), or particular subsets of PASCAL VOC (Everingham *et al.*, 2010) without truncation, occlusion, or "difficult" objects (Aubry *et al.*, 2014).

In this chapter, we aim at joining the two directions, multi-view detection and 3D instance alignment, into *3D object class detection in the wild* – predicting the precise 3D shape and pose of objects of various classes in challenging real world images. We achieve this by combining a robust, part-based object class representation based on RCNNs (Girshick *et al.*, 2014) with a small collection of 3D prototype models, which we align to the observed image at recognition time. The link between a 2D image and a 3D prototype model is established by means of 2D-3D keypoint correspondences, and facilitated by a pose regression step that precedes rigid keypoint alignment.

As a result, the presented method predicts the precise 3D shape and pose of all PASCAL3D+ (Xiang *et al.*, 2014a) classes (Fig. 6.1), at no loss in performance with respect to 2D bounding box localization: our method improves over the previous best results on this dataset (Felzenszwalb *et al.*, 2010) by 21.2% in average precision (AP) while simultaneously improving 12.5% in AAVP (Sect. 6.3.4) in joint object localization and viewpoint estimation (Chapter 5). In addition, projecting the 3D object hypotheses provided by our system onto the image plane result in segmentation masks that are competitive with native segmentation approaches, highlighting the accuracy of our 3D shape estimates.

This chapter makes the following contributions. First, to our knowledge, we present the first method for 3D object class detection in the wild, achieving precise 3D shape and pose estimation at no loss of 2D bounding box localization accuracy compared to state-of-the-art RCNN detectors. Second, we design a four-stage detection pipeline that is explicitly tailored towards 3D object class detection, based on a succession of *(i)* robust 2D object class detection, *(ii)* continuous viewpoint regression, *(iii)* object keypoint detection and *(iv)* 3D lifting through rigid keypoint alignment. Third, we give an in-depth experimental study that validates the design choices at each stage of our system. Crucially, and in contrast to previous work, we demonstrate that enriching the output of the object detector does not incur any performance loss: the final 3D detections yield the same AP as stage (i) and improved AAVP over stage (ii), even though significant geometric detail is added. And fourth, we demonstrate superior performance compared to state-of-the-art in 2D bounding box localization, simultaneous viewpoint estimation, and segmentation based on 3D prototype alignment, on all classes of the PASCAL3D+ dataset (Xiang *et al.*, 2014a).

6.2 3D OBJECT CLASS DETECTION

In this section, we describe our 3D object class detection pipeline. Given a single test image as an input, it can not only predict the 2D bounding box (BB) of each object

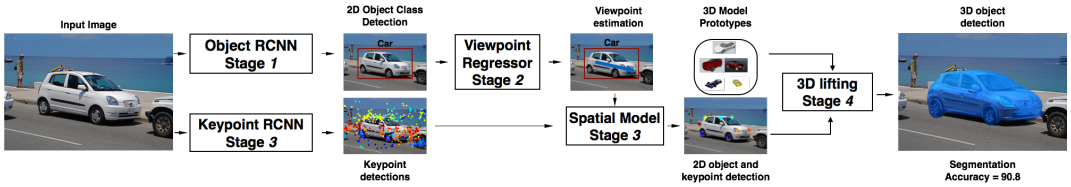


Figure 6.2: Our 3D object class detection pipeline.

in the image, but also yields estimates of their 3D poses as well as their 3D shape, represented relative to a set of prototypical 3D CAD models. Fig. 6.1 gives example results. A schematic overview of our method is shown in Fig. 6.2.

The following subsections provide a walk-through of our pipeline. We start with robust 2D object class detection (Sect. 6.2.1). We then add viewpoint information (Sect. 6.2.2). Next, we localize a set of 3D object keypoints in the 2D image plane (Sect. 6.2.3) that provides the basis for our last stage: 3D lifting (Sect. 6.2.4). It combines all estimates of the previous stages into a final, 3D object class detection result. Since this last step depends crucially on the quality of the intermediate stages, we highlight the important design choices that have to be made in each subsection.

6.2.1 2D Object class detection

RCNNs (Girshick *et al.*, 2014) have shown remarkable performance in image classification and 2D BB localization, leading to state-of-the-art results on the Pascal VOC (Everingham *et al.*, 2010) and ImageNet (Deng *et al.*, 2009) datasets. As precise BB detection and 2D alignment are crucial requirements for being able to infer 3D geometry, we adopt RCNNs as the first stage of our pipeline.

Specifically, we use the implementation of Girshick *et al.* (2014) (RCNN). It consists of three steps: generation of BB proposals, feature extraction using the intermediate layers of a CNN, and subsequent training of a one-vs-all SVMs.

The selective search method (Uijlings *et al.*, 2013) provides several object candidate regions $o \in \mathcal{O}$ in an image. These are passed into a convnet (Krizhevsky *et al.*, 2012) and its unit activations in separate layers are extracted as feature representation for each region. The RCNN uses the responses of either the last convolutional (conv5) or one of the two fully connected layers (fc6, fc7). A SVM is trained for every object class, with the positive examples being the regions with a certain intersection-over-union (IoU) overlap R with the ground truth and the negative examples the regions with IoU ≤ 0.3. At test time, the RCNN provides a set of object detections $o = [o^b, o^c, o^s]$ per image, where o^b is the BB, o^c the object class, and o^s the score.

Empirical results in Girshick *et al.* (2014) on the Pascal VOC 2007 and 2010 datasets identify fc7 features and $R = 1$ as the best set of parameters. We compared the combination of intermediate feature responses and values of R on the Pascal3D+ (Xiang *et al.*, 2014a) dataset and found the same setting to perform best.

6.2.2 Viewpoint estimation

An essential cue for performing the transition from 2D to 3D is an accurate estimate of the 3D pose of the object, or, equivalently, of the viewpoint under which it is imaged. We represent the viewpoint of an object $o^v \in [0, 360)$ in terms of azimuth angle a. Several approaches can be taken to obtain a viewpoint estimate, treating it either as a discrete or continuous quantity. We discuss the discrete version first, mainly to be comparable with recent work. However we argue that due to the continuous nature of the viewpoint the problem should be treated as a continuous regression problem. As the experiments will show (Sect. 6.3.2), this treatment outperforms the discrete variants allowing for a much finer resolution of the viewpoint estimate.

Discrete viewpoint prediction. A large body of previous work and datasets on multi-view object class detection (Savarese and Fei-Fei, 2007; Glasner *et al.*, 2011; Xiang *et al.*, 2014a) use a discretization of the viewpoint into a discrete set of V classes, typically focusing on a single angle (azimuth). The task is then to classify an object hypothesis into one of the $v \in \{1, \ldots, V\}$ classes. While this defeats the continuous nature of the problem, it has the benefit of giving a reduction to a multi-class classification problem for which efficient methods exist.

We conjecture that a convnet representation will be discriminative also for viewpoint estimation and explore two different convnet variants to test this hypothesis. First, we use the pre-trained convnet from Section 6.2.1 and replace the last linear SVM layer for object detection with one for viewpoint estimation. Discretizing the viewpoints in V classes results in V different classifiers for every object category. During test time, we choose the class with the maximum score. We refer to this method as RCNN-MV. We explore a second variant (CNN-MV), a multi-view convnet trained end-to-end to jointly predict category and viewpoint. The convnet parameters are initialized from a network trained on ImageNet (Deng *et al.*, 2009) for object category classification and is then trained using logistic loss and backpropagation (Jia *et al.*, 2014).

Continuous viewpoint prediction. While discrete viewpoint prediction is the de-facto standard today, we believe that angular accurate viewpoint estimation is both more natural and leads to better performance, which is confirmed by the empirical results in Sect. 6.3.2.

We again use the intermediate layer responses of a convnet, pretrained for detection (Section 6.2.1), as the feature representation for this task. From these features, we regress the azimuth angle directly. More formally, let us denote with ϕ_i the features provided by a convnet on region o_i depicting an object of category c. Let o^a represent the azimuth of the region and w^a the azimuth regressor for class c. We use a least squares objective

$$w^a = \underset{w}{\operatorname{argmin}} \, ||o_i^a - \phi_i^\top w||_2^2 + \lambda ||w||_p^2, \tag{6.1}$$

and test three different regularizers: ridge regression ($p = 2$), lasso ($p = 1$), and elastic net. We refer to the regressors as RCNN-Ridge, RCNN-Lasso and RCNN-ElNet. In our experiments, we found that these are the best performing methods,

confirming that the convnet features are informative for viewpoint estimation, and that the continuous nature of the problem should be modeled directly.

6.2.3 Object keypoint detection

While an estimate of the 3D object pose in terms of azimuth angle (Sect. 6.2.2) already conveys significant geometric information beyond a 2D BB, it is not enough to precisely delineate a 3D prototype model, which is the desired final output of our 3D object class detection pipeline. In order to ultimately do the lifting to 3D (Sect.6.2.4), our model relies on additional geometric information in the form of object keypoints. They establish precise correspondences between 3D object coordinates and the 2D image plane.

To that end, we design a set of object class specific keypoint detectors that can accurately localize keypoints in the 2D image plane. In connection with a spatial model spanning multiple keypoints, these detectors can deliver reliable anchor points for 2D-3D lifting.

Keypoints proposal and detection. Recently, it has been shown that powerful part detectors can be obtained by training full-blown object class detectors for parts (Chen *et al.*, 2014). Inspired by these findings, we once more turn to the RCNN as the most powerful object class detector to date, but train it for keypoint detection rather than entire objects. Since keypoints have quite different characteristics in terms of image support and feature statistics, we have to perform the following adjustments to make this work.

First, we find that the standard RCNN mechanism for obtaining candidate regions, selective search (Uijlings *et al.*, 2013), is sub-optimal for our purpose (Sect. 6.3.3), since it provides only limited recall for object keypoints. This is not surprising, since it has been designed to reliably propose regions for entire objects: it starts from a super-pixel segmentation of the test image, which tends to undersegment parts in most cases (Hosang *et al.*, 2014). We hence propose an alternative way of generating candidate regions, by training a separate DPM (Felzenszwalb *et al.*, 2010) detector for each keypoint. To generate positive training examples we need to define a BB around each keypoint. We use a squared region centered at the keypoint that covers 30% of the relative size of the object BB. At test time, we can then choose an appropriate number of candidate keypoint regions by thresholding the DPM's dense sliding window detections.

Second, we find that fine-tuning the convnet on task-specific training data makes a difference for keypoint detection (Sect. 6.3.3). We compare two variants of RCNN keypoint detectors, both scoring DPM keypoint proposal regions using a linear SVM on top of convnet features. The first variant re-uses the convnet features trained for 2D object class detection (Sect. 6.2.1). The second one fine-tunes the convnet on keypoint data prior to feature computation.

Spatial model. Flexible part-based models are among the most successful approaches for object class recognition in numerous incarnations (Fergus *et al.*, 2003; Felzenszwalb and Huttenlocher, 2005; Felzenszwalb *et al.*, 2010), since they constrain

part positions to overall plausible configurations while at the same time being able to adapt to intra-class shape variation – both are crucial traits for the 3D lifting stage of our pipeline. Here, we start from the spatial model suggested by Aubry *et al.* (2014) in the context of localizing mid-level exemplar patches, and extend it for 3D instance alignment. This results in a simple, effective, and computationally efficient spatial model relating object with keypoint detections.

We define a spatial model that relates the position of keypoints to the position of the object center in the 2D image plane, resulting in a star-shaped dependency structure as in previous work (Leibe *et al.*, 2008; Felzenszwalb *et al.*, 2010). Specifically, for every different keypoint class p we estimate on the training data the average relative position around the object center o. Around this estimated mean position we define a rectangular region $N(o, p)$ of size proportional to the standard deviation of the relative keypoint positions in the training set. At test time, for a given object center o, for every part p we perform max-pooling in $N(o, p)$. This prunes out all keypoint detections outside of $N(o, p)$ and only retains the strongest one inside.

As the visibility and relative locations of keypoints changes drastically with object viewpoint, we introduce a number of viewpoint-specific components of this spatial model. During training, these components are obtained by clustering the viewpoints into C clusters, and estimating the mean relative keypoint position on each component.

At test time we resort to two strategies to decide on which component to use. We either use the viewpoint estimation (Sect.6.2.2) as a guidance for which one to use, or we use the one with the best 3D detection objective (Sect. 6.2.4). Indeed, the guided version performs better (Sect. 6.3.3).

6.2.4 3D Object class detection

The result of the previous stages is a combination of a 2D object BB (Sect. 6.2.1) plus a set of 2D keypoint locations (Sect. 6.2.3) specific to the object class. Optionally, the keypoint locations are also specific to viewpoint, by virtue of the viewpoint estimation (Sect. 6.2.2) and the corresponding spatial model component. This input can now be used to lift the 2D object class detection to 3D, resulting in a precise estimate of 3D object shape and pose.

We choose a non-parametric representation of 3D object shape, based on prototypical 3D CAD models for the object class of interest. Assuming known correspondences between keypoints defined on the surface of a particular model and 2D image locations, we can estimate the parameters of the projective transformation that gives rise to the image.

3D Lifting. We adopt the camera model from Xiang *et al.* (2014a) and use a pinhole camera P always facing the center of the world, assuming the object is located there. Assuming a fixed field of view, the camera model consists of 3D rotation (pose) and 3D translation parameters. We parameterize the 3D pose as $o^v \in [0, 360) \times [-90, +90) \times [-180, 180)$, in terms of azimuth angle a, elevation angle e and the in-plane rotation θ. These three continuous parameters, fully specify the

pose of a rigid object. The 3D translation parameters consist of the distance of the object to the camera D and the in-plane translation t.

The 3D lifting procedure jointly estimates the camera and the 3D shape of the object. Let us denote with $\{k^i\}$ the set of 2D keypoint predictions. Let $\{K_j^i\}$ be the corresponding 3D keypoints on the CAD model j and $\tilde{k}_j^i = PK_j^i$ denote the image projection of K_j^i. Then the CAD prototype c^* and camera P^* are obtained by solving

$$c^*, P^* = \underset{c,P}{\operatorname{argmin}} \sum_{i}^{L} ||k^i - \tilde{k}_c^i||. \tag{6.2}$$

We perform exhaustive search over the set of CAD models and solve for P using an interior point solver as in Xiang *et al.* (2014a).

Initialization. The object viewpoint estimate is used to initialize the azimuth. The elevation is initialized using the category mean. We initialize $\theta = 0$. For the in-plane translation and 3D distance parameters, we solve Eq. 6.2 optimizing only for these parameters. This gives a good coarse initialization of the distance and the in-plane translation that is used later for the joint optimization of all parameters.

6.3 EXPERIMENTS

In this section, we give an in-depth experimental study of the performance of our 3D object class detection pipeline, highlighting three distinct aspects. First, we validate the design choices at each stage of our pipeline, 2D object class detection (Sect. 6.3.1), continuous viewpoint regression (Sect. 6.3.2), keypoint detection (Sect. 6.3.3) and 3D lifting (Sect. 6.3.4), ensuring that each stage delivers optimal performance when considered in isolation. Second, we verify that adding geometric detail through adding more pipeline stages does not come at the cost of losing any performance, as it is often observed in previous work (Liebelt and Schmid, 2010; Zia *et al.*, 2013a). And third, we compare the performance of our method to the previous state-of-the-art art, demonstrating significant performance gains in 2D BB localization, simultaneous localization and viewpoint estimation, and segmentation based on 3D prototype alignment. In contrast to previous work (Zia *et al.*, 2013a; Lim *et al.*, 2013; Aubry *et al.*, 2014; Lim *et al.*, 2014), we evaluate the performance of our method for a variety of classes on challenging, real-world images of PASCAL VOC (Everingham *et al.*, 2010; Xiang *et al.*, 2014a).

Dataset. We focus our evaluation on the recently proposed Pascal3D+ (Xiang *et al.*, 2014a) dataset. It enriches PASCAL VOC 2012 (Everingham *et al.*, 2010) with 3D annotations in the form of aligned 3D CAD models. The dataset provides aligned CAD models for 11 rigid classes (*aeroplane*, *bicycle*, *boat*, *bus*, *car*, *chair*, *dining table*, *motorbike*, *sofa*, *train*, and *tv monitor*) of the *train* and *val* subsets of PASCAL VOC 2012. The alignments are obtained through human supervision, by first selecting the visually most similar CAD model for each instance, and specifying the correspondences between a set of 3D CAD model keypoints and their image

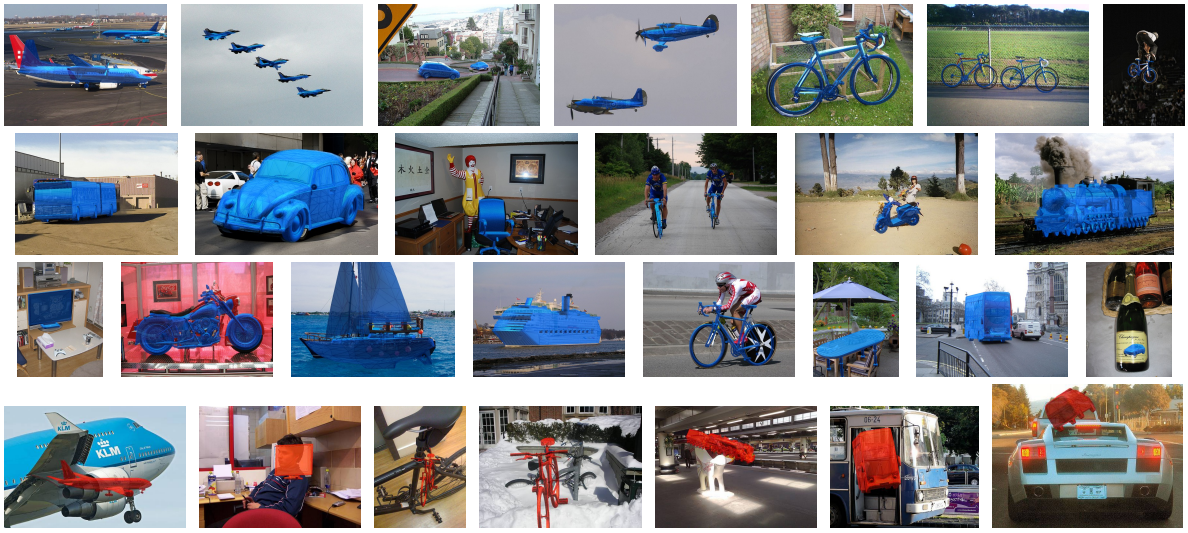


Figure 6.3: 3D CAD prototype alignment examples. (Blue) good alignments, (red) bad alignments. RCNN-Ridge-L fails mainly on truncated and occluded cases.

projections, which are used to compute the 3D pose of the instance in the image. Note that, while the 3D lifting stage of our pipeline (Sect. 6.2.4) is in fact inspired by this procedure, it is entirely automatic, and selects the best fitting 3D CAD model prototype without any human supervision. Throughout the evaluation, we use the *train* set for training and the *val* set for testing, as suggested by the Pascal3D+ (Xiang *et al.*, 2014a).

State-of-the-art. We compare the performance of our pipeline to previous state-of-the-art results on the PASCAL3D+ dataset as reported in Xiang *et al.* (2014a). Specifically, we compare our results to two variants of the deformable part model (DPM (Felzenszwalb *et al.*, 2010)) that predict viewpoint estimates in the form of angular bins in addition to 2D BBs: *(i)* VDPM (Xiang *et al.*, 2014a) trains dedicated mixture components for each angular viewpoint bin, using standard hinge-loss, and *(ii)* DPM-VOC+VP (see Chapter 5). This method has been shown to outperform previous work in multi-view detection by significant margins on 3D Object Classes (Savarese and Fei-Fei, 2007) and PASCAL VOC 2007 cars and bicycles.

6.3.1 2D Bounding box localization

We start by evaluating the first stage of our pipeline, 2D object class detection (Sect. 6.2.1), in the classical 2D BB localization task, as defined by PASCAL VOC (Everingham *et al.*, 2010). Fig 6.4 (left) compares the performance of our RCNN in its discrete multi-view variant RCNN-MV (cyan) to CNN-MV (green) and the state-of-the-art methods on this dataset, VDPM (Xiang *et al.*, 2014a) (blue) and DPM-VOC+VP from Chapter 5 (light blue). It reports the mean average precision (mAP) over all 11 classes of Pascal3D+, for different numbers of discrete azimuth bins, as suggested by the PASCAL3D+ benchmark: VP_1, VP_4, VP_8, VP_{16} and VP_{24} denote the number of

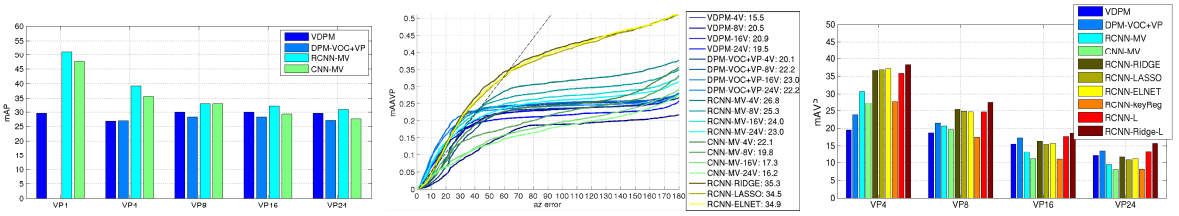


Figure 6.4: (Left) 2D BB localization on Pascal3D+ (Xiang *et al.*, 2014a). (Center, right) Simultaneous 2D BB localization and viewpoint estimation. (Center) continuous mAAVP performance, (right) discrete mAVP performance for VP_4, VP_8, VP_{16} and VP_{24}.

discrete viewpoint-dependent components of the respective model. Note that for the VP_1 case, the VDPM model reduces to the standard DPM (Felzenszwalb *et al.*, 2010) and RCNN-MV to the standard RCNN.

Results. We make the following observations. First, for VP_1, both RCNN (51.2%) and CNN (47.6%) outperform the previous state-of-the-art result of VDPM (29.6%) by significant margins of 21.6% and 18.0%, respectively, in line with prior reports concerning the superiority of CNN- over DPM-based detectors (Girshick *et al.*, 2014). Second, we observe that the performance of VDPM and DPM-VOC+VP remains stable or even slightly increases when increasing the number of components (e.g., from 29.6% to 30.0% for VDPM and from 27.0% to 28.3% for DPM-VOC+VP and VP_{16}). Curiously, this tendency is essentially inverted for RCNN and CNN: performance drops dramatically from 51.2% to 30.8% and from 47.6% to 27.6% for AP_{24}, respectively.

Conclusion. We conclude that, while the training of per-viewpoint components is a viable strategy for DPM-based methods, RCNN-MV and CNN-MV both suffer from the decrease in training data available per component. We hence elect RCNN as the first stage of our 3D detection pipeline, leaving us with the need for another pipeline stage capable of estimating viewpoint.

6.3.2 Simultaneous 2D BB and viewpoint estimation

The original PASCAL3D+ work (Xiang *et al.*, 2014a) suggests to quantify the performance of simultaneous 2D BB localization and viewpoint estimation via a combined measure, average viewpoint precision (AVP). It extends the traditional PASCAL VOC (Everingham *et al.*, 2010) detection criterion to only consider a detection a true positive if it satisfies both the IoU BB overlap criterion *and* correctly predicts the ground truth viewpoint bin (AVP $\leq$ AP). This evaluation is repeated for different numbers of azimuth angle bins VP_4, VP_8, VP_{16} and VP_{24}. While this is a step in the right direction, we believe that viewpoint is inherently a continuous quantity that should be evaluated accordingly. We hence propose to consider the entire continuum of possible azimuth angle errors $D \in [0°, \ldots, 180°]$, and count a detection as a true positive if it satisfies the IoU and is within D degrees of the ground truth.

We then plot a curve over D, and aggregate the result as the average AVP (AAVP). This measure has the advantage that it properly quantifies angular errors rather than equalizing all misclassified detections, and it alleviates the somewhat arbitrary choice of bin centers.

Fig. 6.4 (center) gives the results according to this measure, averaged over all 11 classes of PASCAL3D+. Fig. 6.4 (right) gives the corresponding results in the original AVP measure for discrete azimuth angle binnings (Xiang *et al.*, 2014a) as a reference. In both cases, we compare the performance of our different RCNN-viewpoint regressor combinations, RCNN-Ridge, RCNN-Lasso, and RCNN-ElNet, to the discrete multi-view RCNN-MV and CNN-MV, and the state-of-the-art methods VDPM and DPM-VOC+VP.

Results. We observe that in the mAAVP measure (Fig. 6.4 (left)), the RCNN-viewpoint regressor combinations outperform the previous state-of-the-art methods VDPM and DPM-VOC+VP by large margins. The best performing combination RCNN-Ridge (35.3%, brown) outperforms the best VDPM-16V (20.9%) by 14.4% and the best DPM-VOC+VP-16V (23.0%) by 12.3%, respectively.

The performance of VDPM and DPM-VOC+VP is stable or increasing for increasing numbers of components: VDPM-4V (15.5%) improves to VDPM-16V (20.9%), and DPM-VOC+VP-4 (20.1%) improves to DPM-VOC+VP-16V (23.0%). In contrast, performance decreases for RCNN-MV and CNN-MV: RCNN-MV-4V (26.8%) decreases to RCNN-MV-24V (23.0%), and CNN-MV-4V (22.1%) decreases to CNN-MV-24V (16.2%). Even though the best performing RCNN-MV-4V (26.8%) outperforms the previous state-of-the-art DPM-VOC+VP-16V (23.0%), it can not compete with the RCNN-viewpoint regressor combinations.

The same tendencies are also reflected in the original mAVP measure (Xiang *et al.*, 2014a) (Fig. 6.4 (right)). While DPM-VOC+VP has a slight edge for the fine binnings (it outperforms RCNN-Ridge by 0.9% for VP_{16} and 1.9% for VP_{24}), RCNN-viewpoint regressor combinations dominate for the coarser binnings VP_4 and VP_8, followed by RCNN-MV, CNN-MV, VDPM, and DPM-VOC+VP.

Conclusion. The combination of RCNN and viewpoint regressor RCNN-Ridge provides a pronounced improvement in simultaneous 2D BB localization and viewpoint estimation compared to previous state-of-the-art (12.3% in mAAVP). Notably, it retains the original performance in 2D BB localization of the RCNN (51.2% in AP).

6.3.3 2D Keypoint detection

We proceed by evaluating the basis for our 3D lifting stage, 2D keypoint detection (Sect. 6.2.3), in isolation. We use the keypoint annotations provided as part of Pascal3D+ (Xiang *et al.*, 2014a), and train an RCNN keypoint detector for each of 117 types of keypoints distributed over 11 object categories. Since the keypoints are only characterized by their location (not extent), we evaluate localization performance in a way that is inspired by human body pose estimation (Yang and Ramanan, 2013). For computing a precision-recall curve, we replace the standard BB IoU criterion for

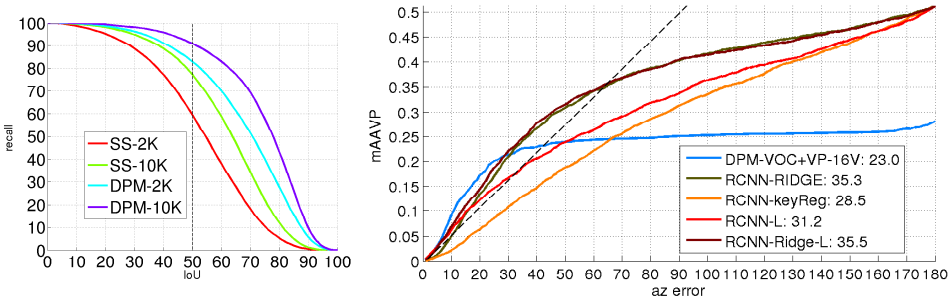


Figure 6.5: Left: 2D Keypoint region proposal quality. Right: Simultaneous 2D BB and viewpoint estimation with 3D lifting.

APP	aero plane	bike	boat	bus	car	chair	din. table	mot. bike	sofa	train	tv	AVG
DPM	19.2	36.2	8.9	26.4	14.3	3.1	4.0	24.2	7.6	8.5	6.1	14.4
RCNN	24.6	43.1	9.8	47.8	34.1	5.7	4.6	36.7	**14.3**	22.5	21.5	24.1
RCNN FT	**30.4**	**48.9**	**12.4**	**50.8**	**39.5**	**9.5**	**6.3**	**41.6**	14.0	**24.5**	**22.8**	**27.3**

Table 6.1: Keypoint detection performance in APP.

detection with an allowed distance P from the keypoint annotation, normalized to a reference object height H. We refer to this measure as Average Pixel Precision (APP). In all experiments, we use $H = 100$ and $P = 25$.

Region proposals. We first evaluate the keypoint region proposal method (Fig. 6.5 (left)), comparing selective search (SS) with the deformable part model (DPM (Felzenszwalb *et al.*, 2010)) at $K = 2000$ and $K = 10000$ top-scoring regions per image. The DPM is trained independently for each keypoint (for that purpose, we define the BB of each keypoint to be a square centered at the keypoint with area equal to 30% of the object area). Both DPM versions outperform the corresponding SS methods by large margin: at 70% IoU DPM with $K = 10000$ gives 30% more recall than SS-10K which is why we stick with these keypoint proposals for our 3D object class detection pipeline.

Part localization. Tab. 6.1 compares the performance of our RCNN keypoint detectors with the DPM keypoint proposal detectors alone, in APP. On average, the RCNN-FT keypoint detectors trained using the features from the CNN fine-tuned on keypoint detection (27.3%) outperform the DPM (14.4%) by 12.9% APP providing a solid basis for our 3D lifting procedure.

6.3.4 2D to 3D lifting

Finally, we evaluate the performance of our full 3D object class detection pipeline that predicts the precise 3D shape and pose. We first give results on simultaneous 2D BB localization and viewpoint estimation as before, but then move on to measuring

sAcc	aero plane	bike	boat	bus	car	chair	din. table	mot. bike	sofa	train	tv	AVG
GT	58.3	32.0	57.9	84.9	79.6	53.5	63.1	69.3	64.7	70.5	80.7	65.0
RCNN-KeyReg	27.1	20.2	19.1	56.2	47.7	23.0	18.6	41.3	**46.4**	30.9	**70.0**	36.4
RCNN-L	30.3	22.0	**27.9**	60.5	44.2	24.9	24.4	46.3	41.9	37.5	45.6	36.9
RCNN-Ridge-L	**35.1**	**22.2**	26.9	**66.4**	**53.9**	**26.8**	**28.6**	**49.0**	44.8	**42.5**	58.7	**41.4**

Table 6.2: Segmentation accuracy on Pascal3D+.

the quality of our predicted 3D shape estimates, in the form of a segmentation task. We generate segmentation masks by simply projecting the predicted 3D shape (Fig. 6.1 (right)). We compare the performance of a direct 3D lifting (RCNN-L) of detected 2D keypoints with a viewpoint guided 3D lifting (RCNN-Ridge-L), and a baseline that regresses keypoint positions (RCNN-KeyReg) on top of an RCNN object detector rather than using keypoint detections.

Simultaneous 2D BB & VP estimation. Fig. 6.5 (right) compares the mAAVP performance of the lifting methods with the best viewpoint regressor RCNN-Ridge and the best previously published method DPM-VOC+VP-16V. Fig. 6.4 (right) gives the AVP_V (Xiang *et al.*, 2014a) performance in comparison with all viewpoint classifiers and regressors.

RCNN-L (31.2% mAAVP) and RCNN-Ridge-L (35.5%) outperform both the RCNN-KeyReg (28.5%) and the DPM-VOC+VP-16V (23.0%) by considerable margins. RCNN-Ridge-L consistently outperforms RCNN-Ridge in terms of AVP_V (by 1.6%, 2.2%, 2.2%, and 4.1% for increasing V), thus improving over the previous pipeline stage. Furthermore, with 18.6% AVP_{16} and 15.8% AVP_{24} it also outperforms DPM-VOC+VP-16V (17.3%,13.6%, respectively), and achieving state-of-the-art simultaneous BB localization and viewpoint estimation results on Pascal3D+.

Segmentation. Tab. 6.2 reports the segmentation accuracy on Pascal3D+. We use the evaluation protocol of Xiang *et al.* (2014a) with two differences. First, we evaluate inside the ground truth BB only to account for truncated and occluded objects. Second, we focus the evaluation on objects with actual ground truth 3D prototype alignment as that constitutes the relevant set of objects we want to compare on. Therefore, we report the performance of the ground truth aligned 3D CAD prototypes (GT) as well.

With 41.4% performance across all classes, RCNN-Ridge-L outperforms RCNN-L (36.9%) and the baseline RCNN-KeyReg (36.4%) by 4%, confirming the quality of the alignment. Fig. 6.3 illustrates successful 2D-3D alignments for different object classes, along with failure cases. Truncated and occluded objects represent a major part of the failures.

In Tab. 6.3 we go one step further and compare to native state-of-the-art segmentation methods (O_2P (Carreira *et al.*, 2012)), this time on the Pascal-context (Mottaghi *et al.*, 2014) dataset. We report the performance on the 11 classes from Pascal3D+ only. RCNN-Ridge-L with 31.5% is only slightly worse than O_2P+ (35.9%) although the latter is designed for segmentation.

sAcc	aero plane	bike	boat	bus	car	chair	din. table	mot. bike	sofa	train	tv	AVG
GT	40.3	27.9	36.2	75.0	59.3	34.9	16.0	59.0	25.2	57.0	72.5	45.7
O_2P	48.2	32.5	29.6	**61.1**	46.7	12.4	12.4	**46.0**	17.0	36.7	41.6	34.9
O_2P+	**52.4**	**32.8**	**33.1**	60.5	**47.8**	12.8	**13.0**	44.5	16.7	**40.1**	40.7	**35.9**
RCNN-KeyReg	21.9	17.2	15.1	49.5	39.2	16.4	11.8	37.3	**21.9**	28.2	**60.9**	29.0
RCNN-L	26.7	18.8	17.5	53.9	36.7	16.2	6.4	43.5	16.3	35.5	49.7	29.2
RCNN-Ridge-L	27.7	20.1	19.9	59.0	41.7	**18.2**	7.8	44.4	18.5	37.9	51.1	31.5

Table 6.3: Segmentation accuracy on Pascal-context (Mottaghi *et al.*, 2014) dataset.

Conclusion. We conclude that RCNN-Ridge-L achieves state-of-the-art simultaneous BB localization and viewpoint estimation performance on Pascal3D+ (Xiang *et al.*, 2014a), outperforming the DPM-VOC+VP and the RCNN-Ridge regressor. It successfully predicts the 3D object shape which is confirmed by it's segmentation performance.

6.4 CONCLUSION

In this chapter, we have build a 3D object class detector, capable of detecting objects of multiple object categories in the wild (Pascal3D+). It consists of four main stages: *(i)* object detection, *(ii)* viewpoint estimation, *(iii)* keypoint detection and *(iv)* 2D-3D lifting. Based on careful design choices, our 3D object class detector improves the performance in each stage, achieving state-of-the-art 3D detection and simultaneous BB localization and viewpoint estimation performance on the challenging Pascal3D+ dataset. At the same time, it accurately predicts the 3D shape of objects, as confirmed by it's segmentation quality. The final result is a rich 3D representation, consisting of 3D shape, 3D viewpoint, and 3D position automatically estimated from only a single image.

7 FINE-GRAINED CATEGORIZATION FOR 3D SCENE UNDERSTANDING

Contents

FINE-GRAINED CATEGORIZATION of object classes is receiving increased attention, since it promises to automate classification tasks that are difficult even for humans, such as the distinction between different animal species, different brand-types of vehicles etc. In this chapter, we consider fine-grained categorization for a different reason. Following the intuition that fine-grained categories encode metric information, we aim to generate metric constraints from fine-grained category predictions, for the benefit of 3D scene-understanding. To that end, while the previous chapters focused on explicit 3D object representations, in this chapter we explore fine-grained representations. Motivated by the fact that part appearance, geometry and part constellations encode subordinate affiliation, we contribute with two part-based fine-grained representations, in addition to a fine-grained dataset of cars. Furthermore, we empirically demonstrate that the richer fine-grained object representation can be further used to localize objects more tightly and accurately in 3D space.

7.1 INTRODUCTION

The recognition of basic-level object categories (Rosch *et al.*, 1976) in natural images has made remarkable progress over the last decade, both in image-level categorization and bounding box localization settings (Everingham *et al.*, 2010). More recently, the recognition of finer-grained, subordinate categories is receiving increased attention (Bar-Hillel and Weinshall, 2006; Nilsback and Zisserman, 2008; Welinder *et al.*, 2010b; Branson *et al.*, 2010; Maji *et al.*, 2011; Farrell *et al.*, 2011; Yao *et al.*, 2011; Wah *et al.*, 2011; Zia *et al.*, 2011). The problem of fine-grained categorization is deemed

challenging due to the need to capture subtle appearance differences between categories while at the same time maintaining robustness to intra-category variations induced by changes in pose and viewpoint. As a consequence, the focus of previous work has been mostly on object categories *and* methods that favor discrimination by strong local appearance cues (such as random color image patches for birds (Yao *et al.*, 2011)) or global image statistics (such as color histograms for flowers (Nilsback and Zisserman, 2008)). In this setting, computer vision techniques could be shown to facilitate fine-grained categorization tasks that are difficult even for humans due to the sheer number and diversity of subordinate categories (Nilsback and Zisserman, 2008; Branson *et al.*, 2010; Wah *et al.*, 2011).

This chapter goes beyond previous work on fine-grained categorization in two ways. First, in addition to exploring the task of fine-grained categorization itself, we suggest the use of fine-grained category predictions as an input for higher-level reasoning. This is based on the observation that fine-grained categories can encode, among other aspects, information about metric object sizes, which can in turn provide geometric constraints for scene-level reasoning. Following this line of argumentation, we focus our attention on rigid, geometric objects that can provide, if correctly categorized, reliable metric size estimates, and introduce a novel dataset of fine-grained car types as a test bed for our approach. This data set is annotated with 2D bounding boxes, viewpoint estimates, car types, and additionally includes metric object sizes (length, width, and height) for use in geometric reasoning.

The second way this chapter departs from previous work (Nilsback and Zisserman, 2008; Yao *et al.*, 2011) is that we design a fine-grained object class representation that captures variations in object shape and geometry rather than appearance, in order to match the object class of interest. To that end, we introduce two different variants of utilizing part detections as indicators of object geometry, of varying complexity. Both are based on one of the best-performing object class detector to date, the deformable part model (DPM, Felzenszwalb *et al.* (2010)). The first variant is based on part detections provided by a pre-trained, generic detector for the object class. Similar in spirit to object bank (Li *et al.*, 2010), it generates features from (part) detector responses by spatial pooling, and feeds them into a classifier for categorization. Relying on existing detectors, this first variant is computationally cheap, and outperforms state-of-the-art classifiers on our data set. The second variant uses the DPM directly for fine-grained categorization, by reformulating it as a structured output prediction problem (see Chapter 3), and directly optimizing a multi-class loss function. While this variant is computationally more demanding, it significantly improves over the first, since part detectors are now directly optimized for the task at hand. It outperforms state-of-the-art classifiers by a large margin.

In summary, this chapter makes the following contributions. i) we introduce a novel data set of fine-grained car types that can serve as a test bed for future research on categorization of geometric objects as well as training data for scene-level reasoning methods based on fine-grained categories. ii) we propose two different variants of utilizing part detections for fine-grained categorization of geometric objects, and demonstrate superior performance compared to the state-of-the-art, and

iii) to our knowledge, we are the first to attempt the application of fine-grained category prediction for the benefit of 3D scene-level reasoning. In particular, we show first results for the task of estimating the depth of objects relative to a calibrated monocular camera based on fine-grained category predictions.

7.2 DEFORMABLE PARTS MODELS FOR FINE-GRAINED RECOGNITION

Our approach to fine-grained categorization is applicable for the wide range of object classes that are characterized by shape and geometry rather than appearance. It follows the intuition that object geometry, and hence, category affiliation, can be encoded in the layout of its constituent parts. We thus design two different models that capture part layout. Both build upon the deformable part model (DPM (Felzenszwalb *et al.*, 2010)), but represent part layout information differently.

7.2.1 Bank of Part Detectors

The basis for our first model is an existing DPM detector for the (basic-level) object class of interest. For example, if the fine-grained task at hand is to distinguish between different car types, the basis for our model is a car detector. While our method could be applied in combination with any detector capable of generating dense response maps of part detections, we chose the DPM since it has proven superior to other detectors for a variety of different object classes, including the rigid that ones we are focusing on (Everingham *et al.*, 2010).

Assuming that the detector has been run on an input image, we propose to form features from the generated part response maps, similar in spirit to object-bank (Li *et al.*, 2010). Note that object-bank uses responses of (massive amounts of) entire object class detectors, lending itself to scene-classification problems that provide enough spatial support in terms of image area. In contrast, we focus on fine-grained classification of individual objects, which are likely to cover only small image regions, and expect to capture more fine-grained information by using responses of individual part detectors. Furthermore, using only part detectors is more efficient in terms of computation, since reasoning about pairwise deformation costs can be spared. Concretely, we compute spatial pyramid (SP) (Lazebnik *et al.*, 2006) representations (1×1, 2×2, 3×3, and 4×4 cells) at different scales over the response maps of all parts, over all components of the DPM. For each SP cell, we memorize min and max responses (pooling), concatenate all values into a single feature vector, and train a linear SVM with L2 loss and regularizer. In the following, we refer to this model as part-bank (PB).

7.2.2 Multi-Class Deformable Part Model

The second model constitutes a proper extension of the DPM (Felzenszwalb *et al.*, 2010), which we implement based on its reformulation as a structured output pre-

diction problem as in Chapter 3. Specifically, we phrase the DPM as a (latent) linear multi-class SVM that can be coherently optimized for the multi-class problem, without the need for a posteriori output coding, such as 1-vs-all or 1-vs-1 schemes (Pandey and Lazebnik, 2011). In the following, we refer to this model as structDPM.

The structDPM is trained from a set $\{x_i, y_i\}$ of images x_i and class labels $y_i \in \{1, ..., K\}$. Similar to Felzenszwalb *et al.* (2010), each class y is represented in the model with a set of n components $\{c^y\}$, where n is a free parameter of the model. The structDPM is the union of components across all classes, $\{c^1\} \cup \{c^2\} \cup \ldots \cup \{c^K\}$. The mapping of training examples to components is latent, with the constraint that for every training example x_i, only components of class y_i can be assigned to it. Each component c is composed of a dedicated root p_c^0 and a set of deformable parts p_c^k, the positions of which are aggregated in latent variables $h = \{p_c^k\} \cup c$, together with the component assignment c. Each part is characterized by a HOG (Dalal and Triggs, 2005) template F_c^k and a spatial deformation cost w.r.t. the root d_c^k. For notational convenience we first stack all model parameters in a single vector for each component c, $\beta_c = \left(F_c^0, F_c^1, \ldots, F_c^n, d_c^1, \ldots, d_c^n, b_c\right)$, where b_c is a bias term, and further into a single vector for an entire model $\beta = (\beta_1, \ldots, \beta_M)$. The features are stacked accordingly: $\Psi(x, y, h) = (\psi_1(x, y, h), \ldots, \psi_M(x, y, h))$, with $\psi_k(x, y, h) = [c = k]\psi(x, y, h)$ ($[\cdot]$ is Iverson bracket notation) being the features computed for component k, where $k \in \{c^y\}$. The vector $\Psi(x, y, h)$ is zero except at the c'th position, i.e., $\langle \beta, \Psi(x, y, h) \rangle = \langle \beta_c, \psi_c(x, y, h) \rangle$. During training, we optimize the following latent structured SVM objective:

$$\begin{aligned} \min_{\beta, \xi \geq 0} \quad & \frac{1}{2}\|\beta\|^2 + C \sum_{i=1}^{N} \xi_i \\ \text{sb.t.} \quad & \forall i, \bar{y} \neq y_i : \max_{h_i} \langle \beta, \Psi(x_i, y_i, h_i) \rangle - \max_{h} \langle \beta, \Psi(x_i, \bar{y}, h) \rangle \geq \Delta(y_i, \bar{y}) - \xi_i \end{aligned}$$

where Δ is a loss function, which we instantiate as $\Delta(y, \bar{y}) = [y \neq \bar{y}]$. For both training and test, we allow the root part to move inside the object bounding box by considering all hypotheses which have an overlap of at least 0.4. At test time, we solve $\text{argmax}_{(y,h)} \langle \beta, \Psi(x, y, h) \rangle$.

7.3 EXPERIMENTS

In the following, we carefully analyze the performance of our models. To that end, we introduce a novel data set of fine-grained *car-types*, and conduct experiments in two different settings: first, we evaluate fine-grained categorization in isolation, as a standard multi-class classification task (Section 7.3.2), comparing to state-of-the-art classifiers. Second, we explore fine-grained categorization in the context of 3D scene understanding, showing promising results of estimating object depth from fine-grained category predictions (Section 7.3.3).

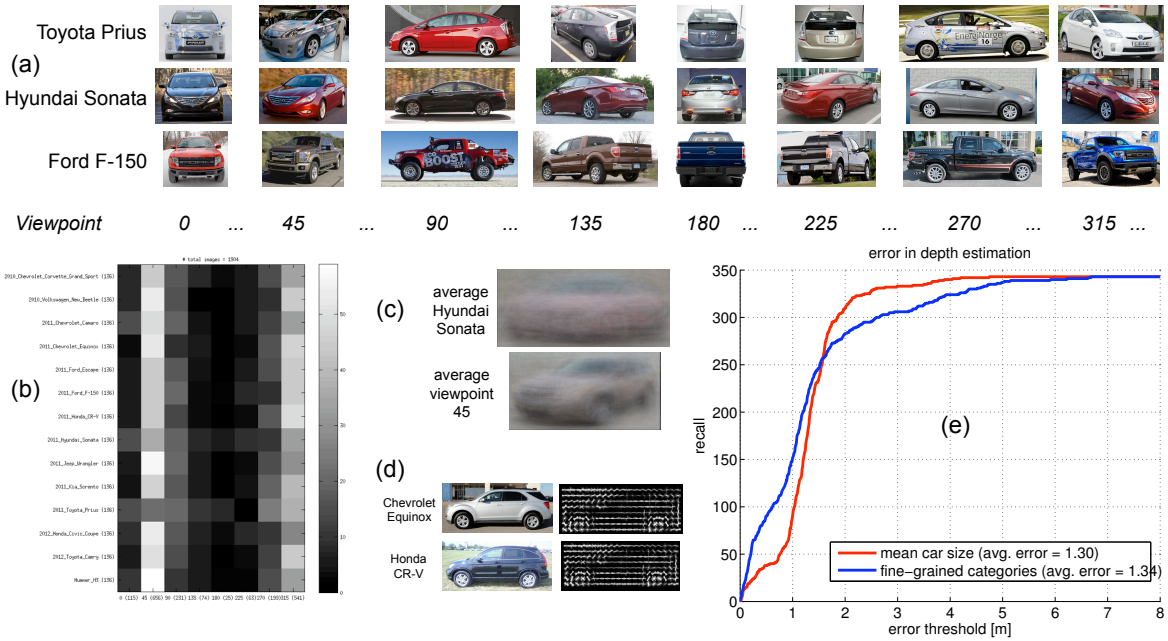


Figure 7.1: Our novel *car-types* data set (Section 7.3.1): (a) example images, (b) statistics, (c) average images, (d) HOG features. (e) Comparison of depth estimation error (Section 7.3.3). This figure is best viewed in the electronic version, with magnification.

7.3.1 Novel Fine-Grained Car Data Set

We introduce a novel data set of fine-grained *car-types*, which we will make publicly available upon publication (Figure 7.1). It consists of 1904 images of cars from 14 different categories (Figure 7.1 (b)), downloaded from the internet. In particular, we queried google image search with terms corresponding to the most frequently appearing sedans, SUVs, sports cars, and compact cars, according to a car trading website. Downloaded images were manually filtered for those that depict at least one car of the queried category in a prominent position. Images are annotated with category labels, 2D bounding boxes, and a viewpoint estimate in the form of the azimuth angle, binned to 5 degrees (we report results on the standard 45 degree binning in the experiments of Section 7.3.2).

Figure 7.1 (a) gives sample images from 3 categories and 8 different viewpoint bins, cropped to approximately the object bounding box. Figure 7.1 (b) gives the number of images for each category and viewpoint, together with the corresponding marginals (in parentheses). We note that the data set is heavily biased w.r.t. viewpoints, which reflects the availability of images we encountered during data collection. It proved almost impossible to collect more than a handful of images for certain combinations of car-type and viewpoint. Figure 7.1 (c) and (d) highlight the challenge of the fine-grained classification problem: images of all categories from a certain viewpoint are better aligned than images from a certain car-type across all

viewpoints (c), and differences in HOG feature space are hard to spot even visually (d). For evaluation, we split the data set into 50% *train*, 25% *val*, and 25% *test* images.

7.3.2 Fine-Grained Categorization

We first evaluate our fine-grained categorization in isolation, as a standard multi-class classification task. We train on the designated *train* data defined by our data set, use *val* for parameter optimization, and test on *test*. For training and test, classifiers are provided images as well as ground truth object bounding boxes, since the task is classification, not detection.

Methods. Tab. 7.1 gives the results for fine-grained categorization on our *car-types* data set, measuring the accuracy of classification as the fraction of correctly classified instances in the test set. It compares four different groups of approaches in its sections, i) baselines, ii) part-bank, iii) structDPM, and iv) combinations of i), ii) and iii). As baselines (i), we consider a HOG (Dalal and Triggs, 2005) template with a linear SVM, and locality constrained linear coding (LLC (Wang *et al.*, 2010b)), which is one of the most powerful image-level classifiers to date (among the state-of-the-art on Caltech-101 (Fei-Fei *et al.*, 2006) and -256 (Griffin *et al.*, 2007) classification benchmarks). For ii), we compare our part-bank (PB) computed on response maps of the DPM (Felzenszwalb *et al.*, 2010) car detector as provided by the authors (Felzenszwalb *et al.*, 2009) (PB(DPM)), and part-bank computed on response maps of the bank of 8 viewpoint-dependent DPMs proposed by (Bao and Savarese, 2011) (PB(mvDPM)). Since the latter explicitly distinguishes between different viewpoints, we expect the corresponding part response maps to be more informative than the ones of PB(DPM). We also add part-bank computed on response maps of our structDPM (PB(structDPM)). For iii), we train our structDPM with 2 components per fine-grained category. For iv), we consider stacking-based combinations of the baselines with the best performing part-bank method PB(mvDPM) (HOG+LLC+PB(mvDPM)) and structDPM (HOG+LLC+structDPM).

Settings. Columns of Tab. 7.1 correspond to different evaluation settings, characterized by the set of class labels provided to the different methods during training: we distinguish *car-type* (col. 1) and both *car-type* and *viewpoint* (col. 2-4). We do the same for testing, which ranges from predicting only *car-type* (col. 1, 14 class problem) to predicting both *car-type* and *viewpoint* (col. 4, 104 classes). Col. 2 and 3 marginalize the predictions of col. 4 over *viewpoint* (col. 2, 14 classes) and *car-type* (col. 3, 8 classes), respectively. Note that the data set does not contain enough images for 8 particular combinations of car-type and viewpoint, which leaves us with 104 classes for the car-type $\times$ vp setting.

Car-type. In Tab. 7.1 col. 1, we observe a clear ordering of performance. While HOG performs moderately (77.5%), it is outperformed by LLC (84.5%) by a large margin (7%). Equally, our PB(DPM) improves over HOG by 6.5%, performing on par

setup		training	car-type	car-type × vp	car-type × vp	car-type × vp
		test	car-type	car-type	vp	car-type × vp
		# categories	14	14	8	104
method	i)	HOG (Dalal and Triggs, 2005)	77.5	81.3	87.8	75.6
		LLC (Wang *et al.*, 2010b)	84.5	82.6	84.2	72.9
	ii)	PB(DPM) *(ours)*	84.0	84.9	88.0	77.1
		PB(mvDPM) *(ours)*	85.3	87.0	88.2	79.4
		PB(structDPM) *(ours)*	89.9	85.5	87.6	77.7
	iii)	structDPM *(ours)*	**93.5**	**88.2**	**88.4**	**79.8**
	iv)	HOG+LLC+PB(mvDPM) *(ours)*	89.1	**88.9**	**89.9**	**81.3**
		HOG+LLC+structDPM *(ours)*	**90.3**	86.3	88.9	79.4

Table 7.1: Comparison of classification accuracy on the *car-types* data set in %, including HOG (Dalal and Triggs, 2005) and LLC (Wang *et al.*, 2010b). Best individual and combined methods are shown in bold font.

with the state-of-the-art LLC. Enriching part-bank with viewpoint information in fact improves performance by 1.3% (PB(mvDPM), 85.3%), and is significantly increased (5.9%) by using parts optimized for the classification problem (PB(structDPM), 89.9%). Using the structDPM end-to-end further increases performance to a striking 93.5%, which is a 9.0% improvement to the best baseline method LLC, and can not be attained by either of the combined methods.

Car-type × vp. In Tab. 7.1 col. 4, we observe a general drop in performance compared to col. 1, due to the increased difficulty of the classification problem (104 vs. 14 classes). The performance of the baselines is reversed – the rigid HOG (75.6%) apparently benefits more from the viewpoint alignment of the training data than LLC (72.9%). Both baselines are consistently outperformed by all variants of part-bank. Again, adding viewpoint information helps (increase from 77.1% for PB(DPM) to 79.4% for PB(mvDPM)). PB(structDPM) performs on par (77.7%). As in col. 1, the best performance for a single method is achieved by structDPM (79.8%), which is remarkable for a 104 class problem. Combining methods improves marginally (to 81.3% for HOG+LLC+PB(mvDPM)).

Marginalizing over *viewpoints* (col. 2), we observe an increase in performance compared to directly predicting the *car-type* (col. 1) for some methods (HOG +3.8%, PB(DPM) +0.9%, PB(mvDPM) +1.7%), and a decrease for others (LLC -1.9%, PB(structDPM) -4.4%, structDPM -5.3%, HOG+LLC+PB(mvDPM) -0.2%, HOG+LLC+structDPM -4.0%).

Marginalizing over *car-types* (col. 3), the performance largely follows the ordering of col. 4. Both baselines (HOG 87.8%, LLC 84.2%) are consistently outperformed by our part-bank classifiers (PB(DPM) 88.0%, PB(mvDPM) 88.2%, PB(structDPM) 87.6%), topped by our structDPM (88.4%) and the combined classifiers (HOG+LLC+PB (mvDPM) 89.9%, HOG+LLC+structDPM 88.9%). In comparison to an existing data set for viewpoint classification into 8 azimuth angle bins (Savarese and Fei-Fei,

2007), where classification is tied to an even more difficult detection setting, the best achieved accuracies on our new data set are considerably worse (89.9% vs. 97.9% by DPM-VOC+VP). This suggests that our data set is also a more challenging test bed for viewpoint classification.

Summary. We conclude that both, part-bank and structDPM, outperform the baselines HOG and LLC by significant margins, in both *car-type* and the even more challenging *car-type* $\times$ *vp* settings. While the computationally more expensive structDPM shows a clear benefit on the former, PB(mvDPM) offers a good compromise between computational efficiency at training time (since it relies on pre-trained detectors) and performance, in particular for the latter setting, where it loses only 0.4% compared to structDPM. Combining methods hardly improves, suggesting that our methods are not complementary to HOG and LLC.

7.3.3 3D Geometric Reasoning

While Section 7.3.2 evaluates our fine-grained categorization in isolation, we now move on to the more challenging task of applying it in the context of a 3D scene understanding task, on a recently proposed street scene data set (Bao and Savarese, 2011; Pandey *et al.*, 2011). To that end, we design an idealized experiment, in which we isolate the contribution of fine-grained category information from possible deficiencies of other system components (such as object localization). While this experiment constitutes a best case evaluation, it highlights that fine-grained category information has the potential to provide valuable constraints in a scene-level reasoning context.

Data set. We use the Ford campus vision and lidar data set (Pandey *et al.*, 2011; Bao and Savarese, 2011) for testing, as it provides calibrated camera images as well as registered point cloud data that can serve as the basis for metric 3D evaluation. Applying fine-grained categorization on this data set is challenging, as its statistics deviate largely from our *car-types* data set used for training, both w.r.t. the imagery (images are taken from an omni-directional camera mounted on a car roof, resulting in image distortions despite correction, and more elevated views of nearby objects) and the objects depicted (cars are not restricted to the types in our data set, and they appear at largely varying, often tiny, scales and are heavily occluded). The data set consists of a number of distinct street scenes, from which we use the test set defined by (Bao and Savarese, 2011), consisting of 141 images in total. Figure 7.2 col. (1) shows examples. We manually annotate the corresponding point clouds with 3D bounding boxes for all visible car objects above a certain size.

Task. We consider the task of predicting the depth of a given object from a single view of the calibrated camera, given its *ground truth* 2D bounding box (which we derive from our 3D annotations), *ground truth* viewpoint (azimuth angle), and its *estimated* physical extent (length, width, and height) as an input. We identify this

distance by casting a ray from the camera center through the center of the 2D object bounding box in the image plane. We then instantiate a 3D bounding box along that ray, which is aligned to the ground plane, sized according to our fine-grain category estimate, and rotated to match the ground truth azimuth, such that the overlap between its 2D projection and the ground truth 2D bounding box is maximized. Maximization is done via exhaustive search over discrete positions on the ray.

Methods. We compare two different methods for estimating the physical extent of an object, which serves as the basis for computing its depth. For the first one, we determine the metric sizes of all *car-types* in our data set (length, width, height) from internet product information. We then apply our fine-grained categorization (structDPM) to all 2D ground truth bounding boxes in the test set, and chose the size of an instantiated 3D object bounding box according to the metric information for the predicted fine-grained category. The second one is our baseline: it ignores fine-grained categories, and instantiates all 3D object bounding boxes with the mean over all metric sizes in our *car-types* data set.

Results. Figure 7.1 (e) gives the results for depth estimation, comparing the performance of using fine-grained category information (blue) with using the mean over all metric sizes (red). It plots the recall of objects with correctly estimated depth according to an error threshold (in meters) vs. that threshold. We observe that using fine-grained category information in fact results in a noticeable improvement in the high precision region of the curve, up to an error of 1.5m (the blue curve stays consistently above the red curve). Beyond that point, the mean over car sizes proves to be more robust than our fine-grained category predictions. This is understandable, given that the test set is quite different from our *car-types* data set used for training, in particular w.r.t. the occurring car-types. Nevertheless, the total average error for fine-grained category predictions is only 4cm larger than for the mean car sizes.

Figure 7.2 visualizes example results. Green arrows highlight improved depth estimates resulting from fine-grained category information, red arrows mark failure cases. In (a), we correctly predict a Ford F150, which is considerably larger than the mean car size, leading to a more accurate depth estimate. (b) shows the same effect with a Chevrolet Corvette Grand Sport. In (c), we correctly predict smaller cars than the mean (Hyundai Sonata and Honda Civic Coupe), also in (d), where we predict a VW New Beetle (which is wrong, but the actual car is small, and can be mistaken for a Beetle). In (e), we mistake the marked car for being an F150, leading to an overestimated size and hence depth.

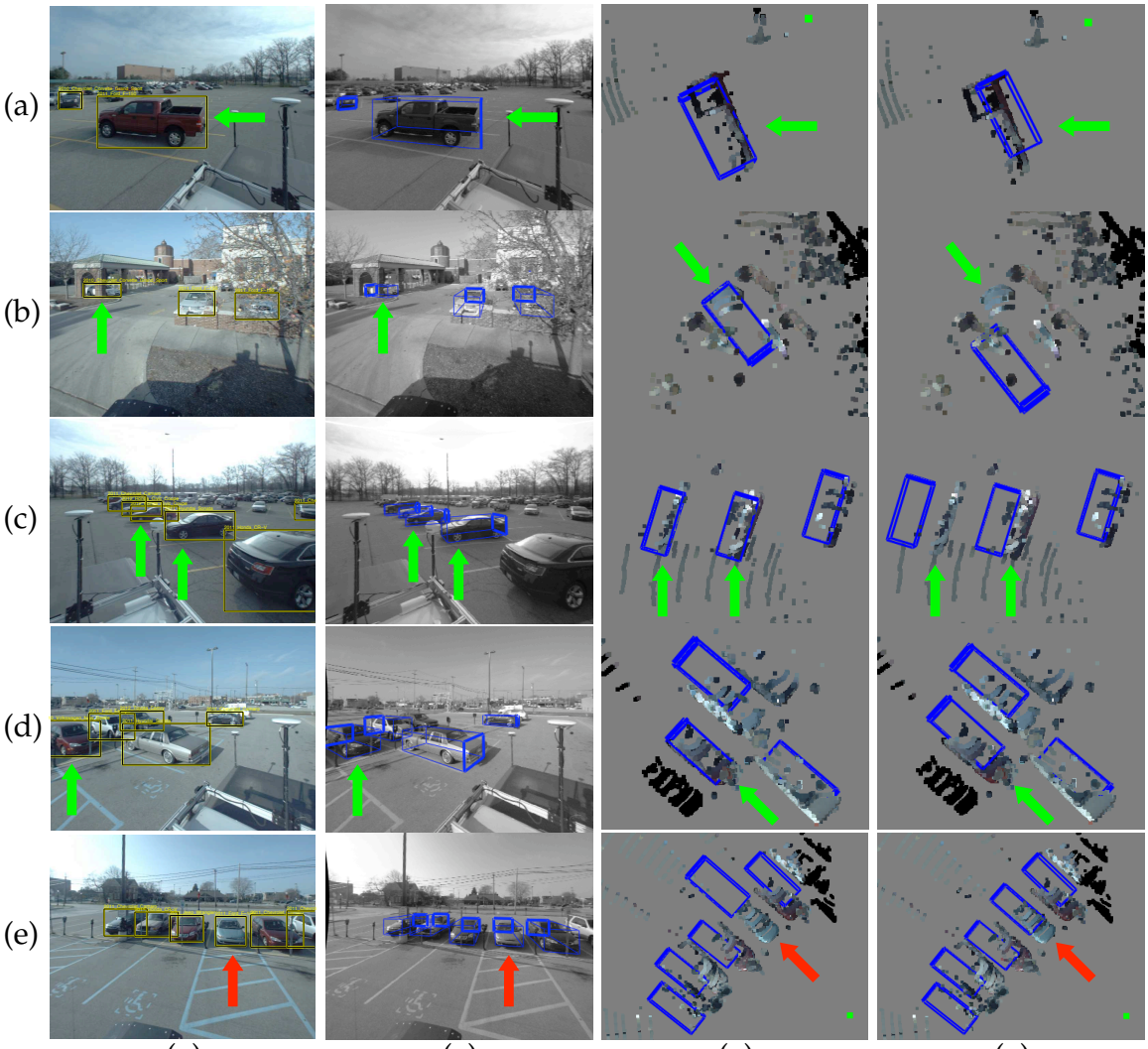


Figure 7.2: Depth estimation results. (1) 2D GT BBs with predicted fine-grained category labels, (2) estimated 3D BBs when using fine-grained category information, (3) point cloud top view for fine-grained, (4) for mean metric sizes. Green: improvement, red: failure. This figure is best viewed in the electronic version, with magnification.

7.4 CONCLUSION

In this chapter, we have considered fine-grained representations of geometric object classes, aiming to use fine-grained category predictions in a 3D scene-understanding context. We introduced two different methods that utilize part detectors to encode category-specific information, which we have shown to outperform baseline classifiers on a newly proposed car-types data set by a significant margin. We further showed first results on using fine-grained category predictions for estimating object depth, which we consider a valuable starting-point for future research.

8 LEARNING MULTI-VIEW PRIORS FROM SPARSE VIEWPOINT DATA

Contents

Fine-grained representations as seen in the previous chapter, have the potential to boost high-level applications like 3D scene understanding. In this chapter, we leverage fine-grained representations in the context of multi-view object detection, driven by the idea that fine-grained representations should result in higher precision in object class detection (e.g. a van is easily confused with an average car, but not that much with a specific car type). We strive towards richer object representation including viewpoints and fine-grained categories and design an object class model that explicitly leverages correlations across visual features in a multi-view knowledge transfer learning framework. Specifically, the presented model in this chapter represents prior distributions over permissible multi-view representations in a parametric way. These multi-view priors are learned once from a source object category and are then subsequently employed in the learning of a target (fine-grained) object category, which might have only sparse training data across viewpoints. We empirically verify that the proposed multi-view knowledge transfer technique can be successfully employed in realistic street-scenes datasets.

8.1 INTRODUCTION

Motivated by higher-level tasks such as scene understanding and object tracking, it has been argued that object class models should not only provide flat, 2D bounding box detections but rather provide more expressive output, such as a viewpoint estimate (Savarese and Fei-Fei, 2007; Su *et al.*, 2009; Liebelt and Schmid, 2010; Stark *et al.*, 2010; Payet and Todorovic, 2011; Villamizar *et al.*, 2011; Glasner *et al.*, 2011) or an estimate of the 3D geometry of the object of interest (Xiang and Savarese, 2012;

Fidler *et al.*, 2012; M.Hejrati and D.Ramanan, 2012; Zia *et al.*, 2013b). Similarly, there has been in increased interest in object representations that allow more fine-grained distinctions than basic-level categories (Lan *et al.*, 2013; Hoai and Zisserman, 2013; Stark *et al.*, 2012), for two reasons. First, these representations potentially perform better in recognition, as they explicitly address the modes of intra-class variation. And second, they, again, can provide further inputs to higher-level reasoning (e.g., in the form of fine-grained category labels).

However, today's methods for 3D and fine-grained object representations suffer from a major weakness: for robust parameter estimation, they tend to require an abundance of annotated training data that covers all relevant aspects (viewpoints, sub-categories) with sufficient density. Unfortunately, this abundance of training data cannot be expected in general. Even in the case of dedicated multi-view datasets (Savarese and Fei-Fei, 2007; Ozuysal *et al.*, 2009; Arie-Nachimson and Basri, 2009; Stark *et al.*, 2012) or when resorting to artificially rendered CAD data (Liebelt and Schmid, 2010; Stark *et al.*, 2010; Zia *et al.*, 2011), the distribution of the number of available training images over object categories is known to be highly unbalanced and heavy-tailed (Wang *et al.*, 2010b; Salakhutdinov *et al.*, 2011; Lim *et al.*, 2011). This is particularly pronounced for categories at finer levels of granularity, such as individual types or brands of cars.

Transfer learning has been acknowledged as a promising way to leverage scarce training data, by reusing once acquired knowledge as a regularizer in novel learning tasks (Fei-Fei *et al.*, 2006; Stark *et al.*, 2009; Gao *et al.*, 2012). While it has been shown that transfer learning can be beneficial for performance, its use in computer vision has, so far, mostly been limited to either classification tasks (Fei-Fei *et al.*, 2006; Zweig and Weinshall, 2007; Rohrbach *et al.*, 2010; Berg *et al.*, 2010) or flat, 2D bounding box detection (Aytar and Zisserman, 2011; Gao *et al.*, 2012), neglecting both the 3D nature of the recognition problem and more fine-grained object class representations.

The starting point and major contribution of this chapter is therefore to design a transfer learning technique that is particularly tailored towards multi-view recognition (encompassing simultaneous bounding box localization and viewpoint estimation). It boosts detector performance for scarce and unbalanced training data, lending itself to fine-grained object representations.

To that end, we represent transferable knowledge as prior distributions over permissible models (Fei-Fei *et al.*, 2006; Gao *et al.*, 2012), in two different flavors. The first flavor (Section 8.2.1) captures sparse correlations between HOG cells in a multi-view deformable part model (DPM, Felzenszwalb *et al.* (2010)), across viewpoints. While this is similar in spirit to Gao *et al.* (2012) in terms of statistical modeling, we explicitly leverage 3D object geometry in order to establish meaningful correspondences between HOG cells, in a fully automatic way. As we show in our experiments (Section 8.3), this already leads to some improvements in performance in comparison to Gao *et al.* (2012). The second flavor (Section 8.2.2) extends the sparse correlations to a full, dense covariance matrix that potentially relates each HOG cell to every other HOG cell, again across viewpoints – this can be seen as directly learning transformations between different views, where the particular choice of source and

target cells can function as a regularizer on the learned transformation, and leads to substantial improvements in simultaneous bounding box localization and viewpoint estimation. Both flavors are simple to implement (covariance computation for prior learning and feature transformation for prior application) and hence widely applicable, yet lead to substantial performance improvements for realistic training data with unbalanced viewpoint distributions.

This chapter makes the following specific contributions: *First*, to our knowledge, our work is the first attempt to explicitly design a transfer learning technique for multi-view recognition and fine-grained object representations. *Second*, we propose two flavors of learning prior distributions over permissible multi-view detectors, one based on sparse correlations between cells and one based on the full covariance. Both are conveniently expressed as instantiations of a class of structured priors that are easy to implement and can be readily applied to current state-of-the-art multi-view detectors (Felzenszwalb *et al.*, 2010). And *third*, we provide an in-depth experimental study of our models, first investigating multi-view transfer learning under the controlled conditions of a standard multi-view benchmark (Savarese and Fei-Fei, 2007) (Section 8.3.1), and finally demonstrating improved performance for simultaneous object localization and viewpoint estimation on realistic street scenes (Geiger *et al.*, 2012) (Section 8.3.2).

8.2 MULTI-VIEW TRANSFER LEARNING

We consider the scenario of transfer learning for object models. The goal is to train an object detection model for a target class for which only very few labeled examples are available. However we have access to an existing (or several) object model for a similar (or the same) object class, the source models. The main intuition that guides our approach is that if we extract common regularities shared by both object classes, then this in turn can be used to devise better target models. In the case of object detection on HOG (Dalal and Triggs, 2005) we reason that although the actual feature distribution may differ, there are similarities in how the features deform under transformations such as viewpoint changes.

Preliminaries. More formally, let us denote by $\mathbf{w}^s$ the parameters of a source model. Specifically in the case of multi component detectors we have $\mathbf{w}^s = [w_1^s, \ldots, w_C^s]$, where the individual w_i^s denote different components of the models. As we are interested in the multi-view setting, the components represent specific viewpoints in our case. Given $\mathbf{w}^s$ and a few N_t labeled examples of a target class $\{x_i, y_i; i \in \{1, \ldots, N_t\}\}$, the goal is to derive a detection model $\mathbf{w}^t$. This is implemented via the regularized risk functional, which has been used for multi-task learning in Evgeniou *et al.* (2005)

$$\mathbf{w}^t = \underset{\mathbf{w}}{\operatorname{argmin}}\, J(\mathbf{w}) + \sum_{i=1}^{N_t} l(\mathbf{w}, x_i, y_i), \tag{8.1}$$

consisting of a regularization $J(\mathbf{w})$ and a data fit term, here the empirical loss l on the training data points. The data term is standard and we may use loss functions such as structured losses or simpler losses like the Hinge loss for classification. In addition to the data term we regularize the model parameters with J, that is derived using information from the source model. We use a transfer learning objective based on Evgeniou *et al.* (2005) where the same regularizer is proposed in the context of multi-task learning. The regularizer is quadratic and of the form

$$J(\mathbf{w}) = \mathbf{w}^\top K_s \mathbf{w}.$$

We distinguish between different choices of K_s, implementing different possibilities to transfer knowledge from the source model. When $K_s = \mathbf{I}$, the objective (8.1) reduces to the standard **SVM** case. In the following, we will explore three different variants for the knowledge transfer matrix K_s.

8.2.1 Learning sparse correlation structures

Let us denote with w an appearance filter of one viewpoint component in the entire set of parameters $\mathbf{w}$. For simplicity we will simply refer to this as w without using sub- or superscripts. This filter is of size $n \times m \times L$. It has spatial extent of $n \times m$ cells, and L are filter values computed in each cell ($L = 32$ in (Felzenszwalb *et al.*, 2010)). We denote each cell j as a vector $w_j \in \mathbb{R}^L$. We implement different versions of the transfer learning objective (8.1) using a graph Laplacian regularization approach ((Evgeniou *et al.*, 2005), Sect 3.1.3) by choosing

$$J(\mathbf{w}) = \mathbf{w}^\top K_s \mathbf{w} = \mathbf{w}^\top (I - \lambda \Sigma_s) \mathbf{w},$$

where Σ_s encodes correlations between different cells in the model. The matrix Σ_s is of size $P \times P$, with P being the total number of model parameters. To distinguish between different choices for the structure of Σ_s we denote with $\sim_n$ a relationship of type n between two cells in w. With "type", for example we can refer to a spatial relation among cells, such as horizontal neighbors, vertical neighbors, etc. This defines a set of cell pairs $\mathbb{P}_n = \{(w_j, w_k) | j \sim_n k\}$ in the model that satisfy relation $\sim_n$. From the set $\mathbb{P}_n$ one can compute cross covariances for different types

$$\Sigma_n = \sum_{j \sim_n k} (w_j - \bar{w})(w_k - \bar{w})^\top, \tag{8.2}$$

where $\bar{w} = \frac{1}{|\mathbb{P}_n|} \sum_j w_j$ is the mean of the set of cells. The full $P \times P$ matrix Σ_s is then constructed from the smaller $L \times L$ block matrices Σ_n (details below). This results in a sparse Σ_s, as the number of cell pairs satisfying a relation is small compared to the total number of possible cell pairs.

Single view correlation structures (SVM-SV). Originally proposed in Gao *et al.* (2012), **SVM-SV** aims at capturing generic neighborhood relationships between HOG cells within a single template (i.e., a single view). This implements a specific

choice for $\sim_n$. **SVM-SV** focuses on 5 specific relation types: horizontal ($\sim_h$), vertical ($\sim_v$), upper-left diagonal ($\sim_{d1}$) and upper-right diagonal ($\sim_{d2}$) cell relations. In addition, **SVM-SV** captures self-correlations of the same cell $\sim_{cell}$. For a given relation type $\sim_n$, **SVM-SV** takes into account all cell pairs in the template which satisfy the relation $\sim_n$, to compute each of the different cross-covariances Σ_h, Σ_v, Σ_{d1}, Σ_{d2} and Σ_{cell}.

Multi-view correlation structures (SVM-MV). We extend **SVM-SV** to encompass multi-view knowledge transfer. In our model different components w of $\mathbf{w}$ correspond to different viewpoints of the object class. Different components are very related since they have a common cause in the geometric structure of the three dimensional object. Therefore, the goal of **SVM-MV** is to capture the across-view cell relations in addition to the single view cell relation introduced by **SVM-SV**. For that purpose, we learn a new, across-view relation type $\sim_{mv}$, capturing cell relationships across different views.

In order to establish cell relationships across viewpoints, we use a 3D CAD model of the object class of interest (or a generic 3D ellipsoid with proper aspect ratio in case we do not have a CAD model available for a class), which provides a unique 3D reference frame across views. The alignment between learned and 3D CAD models is achieved by back-projecting 2D cell positions onto the 3D surface, assuming known viewpoints for the learned models and fixed perspective projection. We then establish cell relationships between cells that back-project onto the same 3D surface patch in neighboring views, and learn Σ_{mv}.

After computing the different cross-covariances Σ_n for both **SVM-SV** and **SVM-MV** from the source models, we construct the Σ_s matrix, which is further on used as a regularizer in the target model training (Eq. 8.1). Σ_s uses the learned cell-cell correlations of different types from the source models, to guide the training of the target model. In order to construct Σ_s, we first establish pairs of cells (w_i, w_j) in the target model which satisfy a certain relation type $\sim_n$ (e.g. neighbors across views) and then we populate the corresponding entries in Σ_s, $\Sigma_s^{i,j}$ with Σ_n. We apply this procedure for all cell relation types defined in **SVM-SV** and **SVM-MV**.

8.2.2 Learning dense multi-view correlation structures (**SVM-Σ**)

SVM-MV and **SVM-SV** capture correlation structures among model cells that satisfy certain cell relations (2D neighboring cells, 3D object surface) resulting in a sparse graph encoded by Σ_s. In the following, we extend this limited structure to a dense graph, that potentially captures relationships among all cells in the model. We will refer to this model as **SVM-Σ**.

Let's assume we are given a set of N source models $\{\mathbf{w}_1^s, \ldots, \mathbf{w}_N^s\}$, for example by training several models using bootstrapping. Then, we compute the un-normalized covariance matrix Σ_{emp}

$$\Sigma_{emp} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{w}_i^s \mathbf{w}_i^{s\top} \tag{8.3}$$

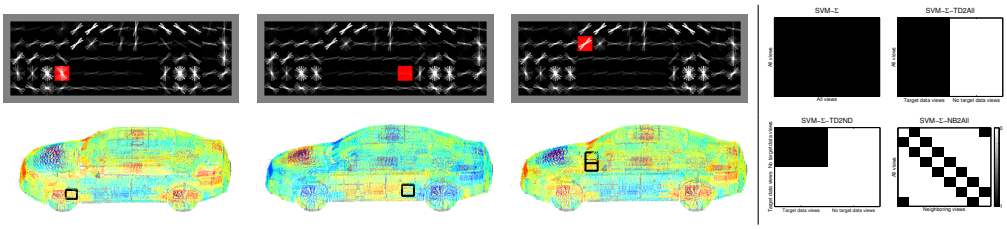


Figure 8.1: (Left) Learned priors visualized in 3D (for a *reference* cell). Red indicates the *reference* cell. The black cube indicates the *reference* cell back-projected into 3D. (Right) **SVM-Σ** versions.

which is a rank N matrix. We set $\Sigma_s = \Sigma_{emp}$.

This variant of Σ_s (**SVM-Σ**) is dense and captures correlations of all types among all cells across all viewpoints in the model. Unlike **SVM-MV** and **SVM-SV**, which are rather generic in nature (e.g., all pairs of horizontal pairs in the template are considered when learning Σ_h), **SVM-Σ** can capture very specific and local cell correlations, within a single template (view) and across views. Figure 8.1 (left) visualizes a heatmap of the strength of the learned correlations for **SVM-Σ** between given reference cells (red squares, black cubes) and all other cells, back-projected onto the 3D surface of a car CAD model. Note that the heatmaps indeed reflect meaningful relationships (e.g., the front wheel surface patch shows high correlation with back wheel patches).

While **SVM-Σ** is a symmetric prior (as the correlations are computed across all views in the model), we also consider the case where the target training data distribution is sparse over viewpoints. We address this by sparser, asymmetric variants of **SVM-Σ** that connect only certain views with each other, by zeroing out parts of the Σ_s using an element-wise multiplication with a sparse matrix S as $S \circ \Sigma_s$. Several choices of S are depicted in Figure 8.1 (right). We distinguish between the following asymmetric priors: **SVM-Σ-TD2ND**, where we transfer knowledge from views for which we have target data to views with no target training data, **SVM-Σ-TD2ALL** with transfer from views with target data to all views, and **SVM-Σ-NB2ALL** where we transfer from every viewpoint to its neighboring viewpoints.

8.2.3 Learning a target model using the learned K_s matrix

We perform model learning (Eq. 8.1) by first doing a Cholesky decomposition of $K_s = U^\top U$. This allows us to define feature and model transformations: $\tilde{\mathbf{x}} = U^{-\top}\mathbf{x}$ and $\tilde{\mathbf{w}} = U\mathbf{w}$. Using these transformations, one can show that $\mathbf{w}^\top K_s \mathbf{w} = \tilde{\mathbf{w}}^\top \tilde{\mathbf{w}}$ and $\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}} = \mathbf{w}^\top \mathbf{x}$, which means we can learn a target model by first, transforming the features and the models using U, then training a model via a standard **SVM** solver in the transformed space, and in the end transforming back the trained model. In the **SVM-SV** case, to be compatible with Gao *et al.* (2012), we perform eigen decomposition instead of Cholesky.

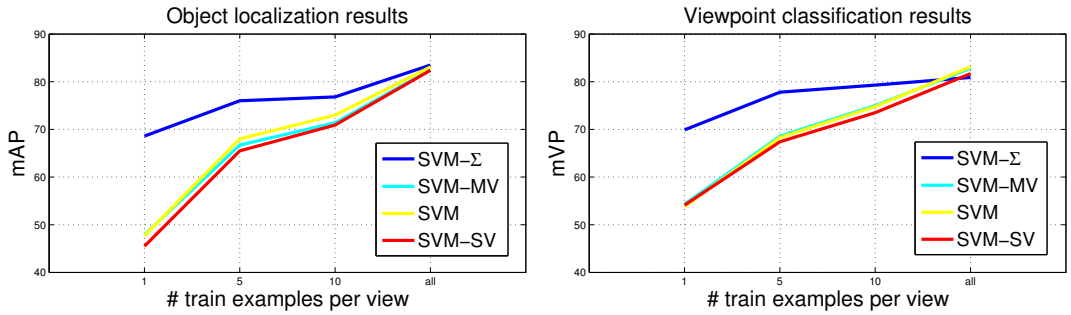


Figure 8.2: 2D BB localization (left) and viewpoint estimation (right) on 3D Object Classes (Savarese and Fei-Fei, 2007).

8.3 EXPERIMENTS

In this section, we carefully evaluate the performance of our multi-view priors. First (Section 8.3.1), we provide an in-depth analysis of different variants of the **SVM-MV** and **SVM-Σ** priors in a controlled training data setting, by varying the viewpoint distribution of the training set. We learn target models using a few target training examples plus our priors and compare them to using the **SVM-SV** prior proposed in Gao *et al.* (2012) and using standard **SVM**. We perform the analysis on two tasks, 2D bounding box localization and viewpoint estimation on the 3D Object Classes dataset (Savarese and Fei-Fei, 2007), demonstrating successful knowledge transfer even for cases in which there is no training data for 3/4 of the viewing circle. Second (Section 8.3.2), we highlight the potential of our **SVM-Σ** priors to greatly improve the performance of simultaneous 2D bounding box localization and viewpoint estimation in a realistic, uncontrolled data set of challenging street scenes (the tracking benchmark of KITTI (Geiger *et al.*, 2012)).

For computational reasons, we restrict ourselves to the root-template-only version of the DPM (Felzenszwalb *et al.*, 2009) as the basis for all our models in Section 8.3.1, but consider the full, part-based version for the more challenging and realistic experiments in Section 8.3.2. In all cases, the C parameter is fixed to 0.002 (Felzenszwalb *et al.*, 2010) for all tested methods. We set $\lambda = 0.9/e_{max}$, where e_{max} is the biggest eigenvalue of Σ_s. We empirically verified that this always resulted in a positive definite matrix K_s, which makes Eq. 8.1 a convex optimization problem.

8.3.1 Comparison of multi-view priors

We start by comparing the different multi-view priors on the 3D Object Classes dataset (Savarese and Fei-Fei, 2007) (a widely accepted multiview-benchmark with balanced training and test data from 8 viewpoint bins, for 9 object classes), in two sets of experiments. In the first set, we use the same number k of target training examples per view (multi-view *k-shot* learning). In the second set, we exclude certain viewpoints completely from the training data ($k = 0$), keeping only a single example

AP/MPPE	car	bicycle	iron	cell	mouse	shoe	stapler	toaster	mAP
SVM-SV	99.6/92.9	88.8/87.6	94.9/94.7	51.0/82.2	61.3/74.7	93.9/81.4	71.5/69.2	94.4/70.6	81.9/81.7
SVM-MV	99.8/95.0	89.9/87.6	96.4/96.1	51.2/82.3	61.2/70.8	93.4/87.3	72.6/70.2	95.3/72.8	82.5/82.8
SVM-Σ	99.8/92.5	96.7/92.2	90.6/88.8	53.7/80.9	61.4/69.8	94.7/86.2	74.2/70.2	97.2/66.7	83.5/80.9
SVM	99.8/95.0	90.1/87.9	97.0/95.5	51.1/81.5	62.5/72.5	94.8/86.5	74.2/70.2	95.2/75.6	83.1/83.1
Lopez-Sastre *et al.*	96.0/89.0	91.0/88.0	53.0/-	43.0/-	41.0/-	78.0/-	32.0/-	54.0/-	61.0/79.2
Gu and Ren	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/74.2
DPM-Hinge +VP	99.6/92.5	98.6/93.0	93.3/86.3	62.9/65.4	73.1/62.2	97.9/71.0	84.4/62.8	96.0/50.0	88.2/72.9
DPM-VOC+VP	99.8/97.5	98.8/97.5	96.0/89.7	62.4/83.0	72.7/76.3	96.9/89.8	83.7/81.2	97.8/79.7	88.5/86.8
Liebelt and Schmid	76.7/70.0	69.8/75.5	-/-	-/-	-/-	-/-	-/-	-/-	-/-
Zia *et al.*	90.4/84.0	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-
Payet and Todorovic	-/86.1	-/80.8	-/-	-/-	-/-	-/-	-/-	-/-	-/-
Glasner *et al.*	99.2/85.3	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-

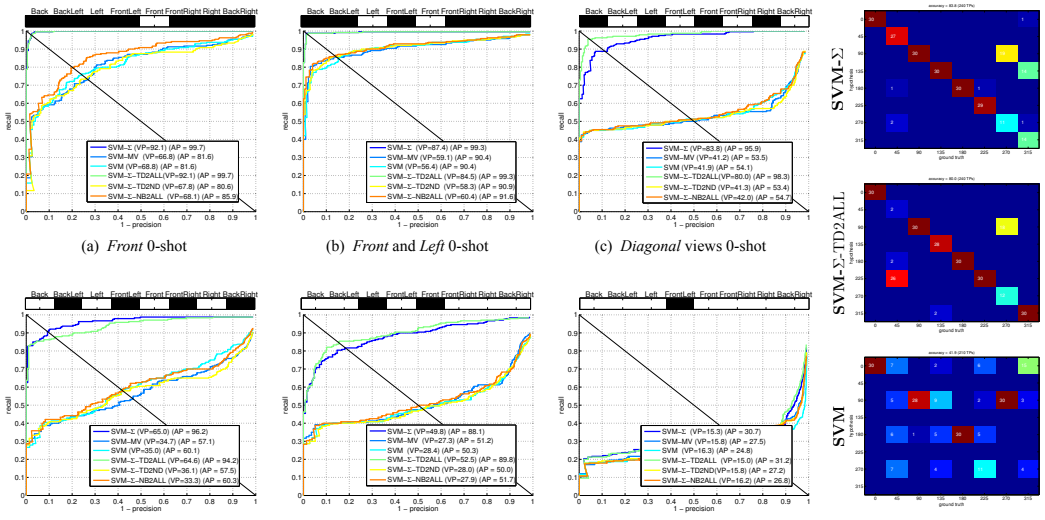
Table 8.1: Comparison to state-of-the-art on 3D Object Classes (Savarese and Fei-Fei, 2007).

from each of the other viewpoints (sparse multi-view *k-shot* learning). In both cases, the test set requires detecting objects seen *from the entire viewing circle*. For each class, our priors are trained using bootstrapping, from 5 source models (each trained from 15 randomly sampled examples per view). The final target model for a class is obtained by using k training examples from that class plus the respective prior.

Multi-view *k-shot* learning. Figure 8.2 plots 2D BB localization (left) and viewpoint estimation (right) performance for **SVM**, **SVM-SV**, **SVM-MV**, and **SVM-Σ**, varying the number $k \in \{1, 5, 10, all\}$ of target training examples per view, averaged over 5 randomized runs. We make the following observations. First, we see that **SVM-Σ** outperforms all other methods by significant margins for restricted training data ($k \in \{1, 5, 10\}$), for both 2D BB localization (by at least 20.1%, 8.0% and 3.8%, respectively) and viewpoint estimation (by 15.6%, 9.2% and 4.3%). Second, the benefit of **SVM-Σ** increases with decreasing number of training examples, saturating for $k = all$. And third, **SVM-SV** (Gao *et al.*, 2012) and **SVM-MV** apparently fail to convey viewpoint-related information beyond what can be learned from the k target examples alone, performing on par with **SVM**.

As a sanity check, Tab. 8.1 relates the complete per-class results for all methods and $k = all$ (rightmost curve points in Figure 8.2) to the state-of-the-art. Despite not using parts, our models in fact outperform previously reported results (Gu and Ren, 2010; Liebelt and Schmid, 2010; Zia *et al.*, 2011; Lopez-Sastre *et al.*, 2011; Payet and Todorovic, 2011; Glasner *et al.*, 2011) with and without priors, except for DPM-VOC+VP based on DPM (Felzenszwalb *et al.*, 2010) with parts. As parts obviously improve performance, we add them in Section 8.3.2.

Sparse multi-view *k-shot* learning. We move on to a more challenging setting in which (single) training examples are only available for selected views, but not for others. Successful localization and viewpoint estimation thus depends on prior information that can be "filled in" for the missing viewpoints. Figure 8.3 plots precision-recall curves for the car class and six different settings of increasing difficulty, not having training data for just one view (front) (a), not for two views (front and left) (b), not for four views (diagonal) (c), off-diagonal (d), not for six views (diagonal, back and right) (e), and not for all views except front-left (f). Average



(a) *Front* 0-shot (b) *Front* and *Left* 0-shot (c) *Diagonal* views 0-shot

(d) *Off-diagonal* views 0-shot (e) Diagonal, *Back* and *Right* 0-shot (f) *All* views except *FrontLeft* 0-shot (g) Viewpoint confusion matrices

Figure 8.3: 3D Object Classes (Savarese and Fei-Fei, 2007). Unbalanced multi-view *o-shot* experiments (on *cars*) with no training data for (a) *Front*, (b) *Front* and Left, (c) *Off-diagonal* views, (d) *Diagonal* views, (e) 1 training example for Left and *Front* views, (f) 1 example for *Front-left* view. (g) VP confusion matrices for the o-shot *Diagonal* case. Bars on top indicate (with black) which viewpoints are used in training for each experiment.

precision and viewpoint estimation results are given in plot legends. We compare the performance of **SVM**, **SVM-MV**, **SVM-Σ**, and three further variations of **SVM-Σ** that restrict the structure of the prior covariance matrix (Section 8.2.2), namely, **SVM-Σ**-TD2ALL, **SVM-Σ**-TD2ND, and **SVM-Σ**-NB2ALL.

In Figure 8.3 (a) to (f), we observe that two methods succeed in transferring information to up to 6 unseen viewpoints (**SVM-Σ**, dark blue, and **SVM-Σ**-TD2ALL, green), with APs ranging from an impressive 99.7 to 88.1% and VPs ranging from 92.1 to 49.8% for **SVM-Σ**). This observation is confirmed by the confusion matrices in Figure 8.3 (g): both **SVM-Σ** and **SVM-Σ**-TD2ALL exhibit a much stronger diagonal structure than **SVM**. Understandably, performance deteriorates for just one observable viewpoint (Figure 8.3 (f); AP drops to 30.7%, VP to 15.3%). **SVM-MV** (light blue) provides an advantage over **SVM** (cyan) only for extremely little data (Figure 8.3 (e), (f)), where it improves AP by 0.9% and 2.7%.

Summary. We conclude that different kinds of priors (**SVM-SV**, **SVM-MV**, and variations of **SVM-Σ**) vary drastically in their ability to convey viewpoint-related information. Notably, we observe only minor differences between **SVM-SV**, **SVM-MV**, and **SVM**, but large gains in both 2D bounding box localization and viewpoint estimation for **SVM-Σ**.

prior		AP / VP			AP+VP-D / AP+VP-C		
method	dataset	*base*	*car-type*	*car-model*	*base*	*car-type*	*car-model*
SVM (KITTI+ 3D obj.)		87.1 / 69.3	- / -	- / -	53.6 / 67.0	- / -	- / -
SVM (KITTI)		86.6 / 68.7	88.7 / 70.9	83.3 / 62.0	53.3 / 65.8	58.1 / 67.9	40.3 / 55.1
SVM-Σ	3D objects	90.7 / 71.9	**91.6** / 75.1	87.5 / 73.9	61.5 / 70.1	65.2 / **74.1**	60.9 / 70.7
SVM-Σ	KITTI	90.7 / 71.9	90.1 / 75.1	89.4 / **75.6**	61.6 / 70.2	**66.1** / 73.5	65.2 / 73.4
SVM-MV	3D objects	90.2 / 72.6	90.3 / 71.9	82.9 / 63.2	60.9 / 69.9	60.8 / 70.0	41.5 / 55.8
SVM-MV	KITTI	89.2 / 73.1	88.5 / 71.1	76.5 / 66.5	62.1 / 69.8	58.8 / 67.6	44.8 / 53.5
SVM-SV	3D objects	90.7 / 71.9	86.5 / 70.4	76.6 / 65.8	61.5 / 70.1	55.9 / 65.8	44.3 / 53.1
SVM-SV	KITTI	86.9 / 71.4	85.8 / 70.8	76.5 / 66.5	59.6 / 67.0	56.5 / 65.1	44.8 / 53.5

Table 8.2: Multi-view detection results on KITTI (Geiger *et al.*, 2012).

8.3.2 Leveraging multi-view priors for object detection

Having verified the ability of our **SVM-Σ** priors to transfer viewpoint information for scarce and unbalanced training data in Section 8.3.1, we now move on to an actual, realistic setting, which naturally exhibits the dataset statistics that we simulated earlier (see Figure 8.4, 8.5 and 8.6). Specifically, we focus on the *car* object class on the KITTI street scene dataset (Geiger *et al.*, 2012) (and the tracking benchmark subset), consisting of 21 sequences (8,008 images, corresponding to 579 car tracks and 27,300 annotated car bounding boxes) taken from a driving vehicle. We use 5 sequences for training and the rest for testing. Due to the car-mounted camera setup, the distribution of viewpoints for car objects is already heavily skewed towards back and diagonal views (cars driving in front of the camera car or being parked on the side of the road, (see Figure 8.4). This becomes even more severe when considering more fine-grained categories, such as individual *car-types* (we distinguish and annotate 7: stat. wagon, convertible, coupe, hatchback, minibus, sedan, suv) and *car-models* (23 in total)

Evaluation criteria. Average precision (AP) computed using the Pascal VOC (Everingham *et al.*, 2006) overlap criterion, based solely on bounding boxes (BB) has been widely used as an evaluation measure. Since the ultimate goal of our approach is to enable simultaneous object localization and viewpoint estimation (both are equally important in an autonomous driving scenario), and in line with Geiger *et al.* (2012), we report performance for two combined measures (jointly addressing both tasks) in addition to AP and VP. Specifically, AP+VP-D allows a detection $\hat{y}$ to be a true positive detection if and only if the viewpoint estimate $\hat{y}^v$ is the same as the ground truth y^v. The second measure, AP+VP-C assigns a weight $\hat{w} = (180^\circ - |\angle(\hat{y}^v, y^v)|)/180^\circ$ to the true positive detection based on how well it aligns with the ground truth viewpoint. In line with Geiger *et al.* (2012), we report results for non-occluded objects.

Basic-level category transfer. We commence by applying our priors to a standard object class detector setup, in which a detector is trained such that positive examples are annotated on the level of basic-level categories (i.e., *car*), denoted *base* in the following. Tab. 8.2 (left) gives the corresponding 2D bounding box localization

and viewpoint estimation results, comparing our priors **SVM-MV** and **SVM-Σ** to **SVM-SV** and a baseline not using any prior (**SVM**). For each, we consider two variants depending from which data the prior (or the detector itself for **SVM**) has been trained (KITTI, 3D Object Classes, or both). Note that the respective prior and **SVM** variants use the exact same training data (but in different ways) and are hence directly comparable in terms of performance.

In Tab. 8.2 (left, col. *base*), we observe that our priors **SVM-MV** and **SVM-Σ** consistently outperform **SVM**, for both 2D BB localization and viewpoint estimation, for both choices of training data (e.g., **SVM-Σ**-KITTI with 90.7% AP and 71.9% VP vs. **SVM**-KITTI with 86.6% AP and 68.7% VP). The performance difference is even more pronounced when considering the combined performance measures (Tab. 8.2 (right, col. *base*)). **SVM-Σ**-KITTI achieves 61.6% AP+VP-D and 70.2% AP+VP-C, outperforming **SVM**-KITTI (53.3%, 65.8%) by a significant margin.

Similarly, **SVM-Σ**-3D Object Classes outperforms **SVM**-KITTI+3D Object Classes in all measures (90.7% vs. 87.1% AP, 71.9% vs. 69.3% VP, 61.5% vs. 53.6% AP+VP-D and 70.1 vs. 67.0% AP+VP-C). **SVM-MV** and **SVM-SV** priors also show promising detection performance, outperforming the **SVM** models in all metrics.

Fine-grained category transfer. Recently, is has been shown that fine-grained object class representations on the level of sub-categories can improve performance (Lan *et al.*, 2013; Hoai and Zisserman, 2013; Stark *et al.*, 2012), since they better capture the different modes of intra-class variation than representations that equalize training examples on the level of basic-level categories. Further, these representations lend themselves to generate additional output in the form of fine-grained category labels that can be useful for higher-level tasks, such as scene understanding. In the following, we hence consider two fine-grained object class representations that decompose *cars* into distinct *car-types* or even individual *car-models*. Both are implemented as a bank of multi-view detectors (one per fine-grained category) that are trained independently, but combined at test time by a joint non-maxima suppression to yield basic-level category detections.

Note that the individual fine-grained detectors suffer even more severely from scarce and unbalanced training data (see Figure 8.5 and 8.6) than on the basic-level (see Figure 8.4) – this is where our priors come into play: we train the priors, as before, on the *base* level, and use them to facilitate the learning of each individual fine-grained detector, effectively transferring knowledge from *base* to fine-grained categories.

Tab. 8.2 gives the corresponding results in columns *car-type* and *car-model*, respectively. We observe: first, performance can in fact improve as a result of the more fine-grained representation, for both **SVM-MV**, **SVM-Σ** and even **SVM** (**SVM**-KITTI-car-type improves AP from 86.6% to 88.7%, and VP from 68.7% to 70.9%, and AP+VP-D from 53.3% to 58.1% and AP+VP-C from 65.8% to 67.9% compared to **SVM**-KITTI-base). A similar boost in performance in viewpoint estimation and combined can be seen for **SVM-Σ** (**SVM-Σ**-KITTI-car-type improves VP from 71.9% to 75.1%, and AP+VP-D from 61.6% to 66.1% and AP+VP-C from 70.2% to 73.5% compared to **SVM-Σ**-KITTI-base; the AP stays consistently high with 90.7% vs.

		@50 iou				@70 iou			
		AP / VP		AP+VP-D / AP+VP-C		AP / VP		AP+VP-D / AP+VP-C	
prior	dataset	*base*	*car-type*	*base*	*car-type*	*base*	*car-type*	*base*	*car-type*
SVM (KITTI)		90.9 / 74.3	93.2 / 75.9	65.2 / 72.1	66.8 / 74.9	49.9 / 74.2	60.0 / 76.8	37.5 / 40.4	44.6 / 48.3
SVM-Σ	3D obj.	**94.8** / 78.6	93.4 / **81.7**	72.1 / 78.7	**73.0 / 80.6**	51.5 / 81.2	**64.7 / 83.9**	41.9 / 44.3	**53.0 / 56.7**
SVM-Σ	KITTI	94.8 / 77.2	94.3 / 78.3	70.4 / 77.3	70.4 / 79.6	49.7 / 79.0	61.2 / 80.4	39.5 / 41.8	47.9 / 53.3

Table 8.3: Multi-view detection results on KITTI (Geiger *et al.*, 2012). Models have root and 4 parts per view.

	without parts							with parts		
prior	**SVM-Σ**	**SVM-Σ**	**SVM-MV**	**SVM-MV**	**SVM-SV**	**SVM-SV**	**SVM**	**SVM-Σ**	**SVM-Σ**	**SVM**
	KITTI	3D obj.	KITTI	3D obj.	KITTI	3D obj.	-	KITTI	3D obj.	-
station wagon	**71.2**	70.2	64.5	63.6	62.6	61.9	61.9	**82.7**	81.9	79.0
convertible	**24.4**	24.0	12.9	10.8	13.8	11.7	12.7	**50.7**	36.8	12.0
coupe	**67.5**	67.1	63.7	67.0	60.5	57.7	67.1	**79.9**	76.6	76.5
hatchback	**89.8**	85.7	66.4	78.2	58.9	65.0	71.0	**95.5**	88.0	87.2
minibus	**31.3**	16.8	20.0	18.7	16.3	18.0	18.6	**59.7**	42.0	41.4
sedan	**69.4**	53.8	46.7	49.4	37.8	41.8	48.7	**83.8**	79.8	66.2
suv	**19.7**	14.7	8.1	7.3	5.2	8.0	8.6	34.5	**35.1**	16.4
mAP	**53.3**	47.5	40.3	42.1	36.4	37.7	41.2	**69.5**	62.9	54.1

Table 8.4: *Car-type* detection results on the KITTI (Geiger *et al.*, 2012) dataset.

90.1%). Second, the level of granularity can be too fine: for almost all methods, the performance of the fine-grained *car-model* drops below the performance of the corresponding *base* detector – there is just so little training data for each of the car models that reliable fine-grained detectors can hardly be learned. Curiously, **SVM-**Σ-KITTI-car-model can still keep up in terms of localization (89.4% AP) and even obtains the overall best VP accuracy of 75.6%, which is also reflected in the combined measures (65.2% AP+VP-D, 73.4% AP+VP-C). Third, **SVM-**Σ-KITTI-car-type is the overall best method, outperforming the original baseline **SVM**-KITTI-base by impressive margins, in particular for the combined measures (90.1% vs. 86.6% AP, 75.1% vs. 68.7% VP, 66.1% vs. 53.3% AP+VP-D, 73.5% vs. 65.8% AP+VP-C).

Tab. 8.3 (left) gives the results for the best performing priors of Tab. 8.2 (**SVM-**Σ-KITTI, **SVM-**Σ-3D Object Classes) in comparison to **SVM**-KITTI, now using parts. As expected, parts result in a general performance boost for all methods (around 5% for all measures). The benefit of our priors remains, for both granularity levels *base* and *car-type*, in particular for the combined measures: **SVM-**Σ-KITTI-base outperforms **SVM**-KITTI-base by similarly large margins as for the no-parts case (70.4% vs. 65.2% AP+VP-D, 77.3% vs. 72.1% AP+VP-C), and **SVM-**Σ-KITTI-car-type outperforms **SVM**-KITTI-car-types by (70.4% vs. 66.8% AP+VP-D, 79.6% vs. 74.9% AP+VP-C).

Tab. 8.3 (right) applies a tighter overlap criterion for true positive detections (0.7 intersection over union) (Geiger *et al.*, 2012). Interestingly, this leads to a larger separation in performance between *base* and *car-type* models, in particular in AP: e.g., **SVM-**Σ-3D Object Classes improves from 51.5% to 64.7%, **SVM-**Σ-KITTI from 49.7% to 61.2% and **SVM**-KITTI improves from 49.9% to 60.0%, highlighting the benefit of the fine-grained object class representation in particular for highly precise detection.

Lastly, we evaluate the performance of our fine-grained detectors on the level

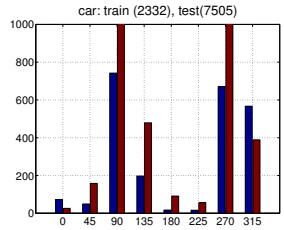


Figure 8.4: *Car* train (blue) and test (red) statistics over 8 viewpoint bins.

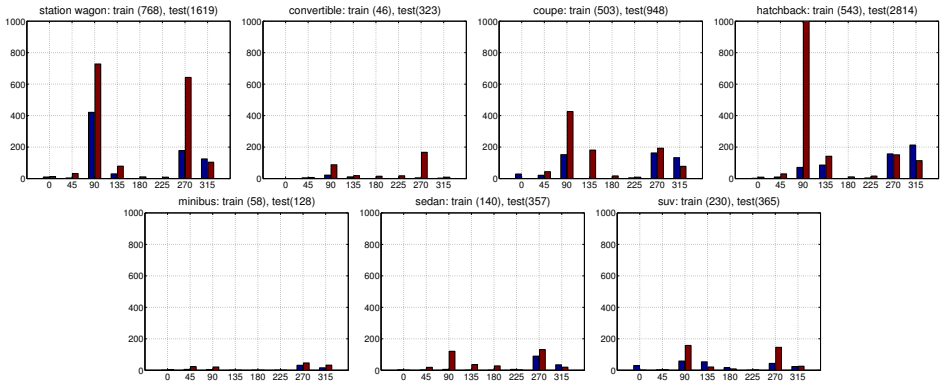


Figure 8.5: *Car-types* train (blue) and test (red) statistics over 8 viewpoint bins.

of the respective fine-grained categories (*car-types*), as independent detection tasks. Tab. 8.4 gives the results without (left) and with parts (right). Again, our priors **SVM-Σ** consistently outperform the baseline **SVM** for all individual categories as well as on average by large margins (53.3% vs. 41.2% mAP for **SVM-Σ**-KITTI without parts, and 69.5% vs. 54.1% with parts).

Summary. We conclude that our priors (in particular **SVM-Σ**) in fact improve performance for simultaneous 2D bounding box localization and viewpoint estimation, for different levels of granularity of the underlying object representation (*base*, *car-type*, *car-model*). Notably, our priors allow for robust learning even on the most fine-grained level of *car-models*, where training data is scarce and unbalanced and **SVM** fails. The combination of fine-grained representation and prior results in a pronounced performance gain compared to **SVM** on the *base* level.

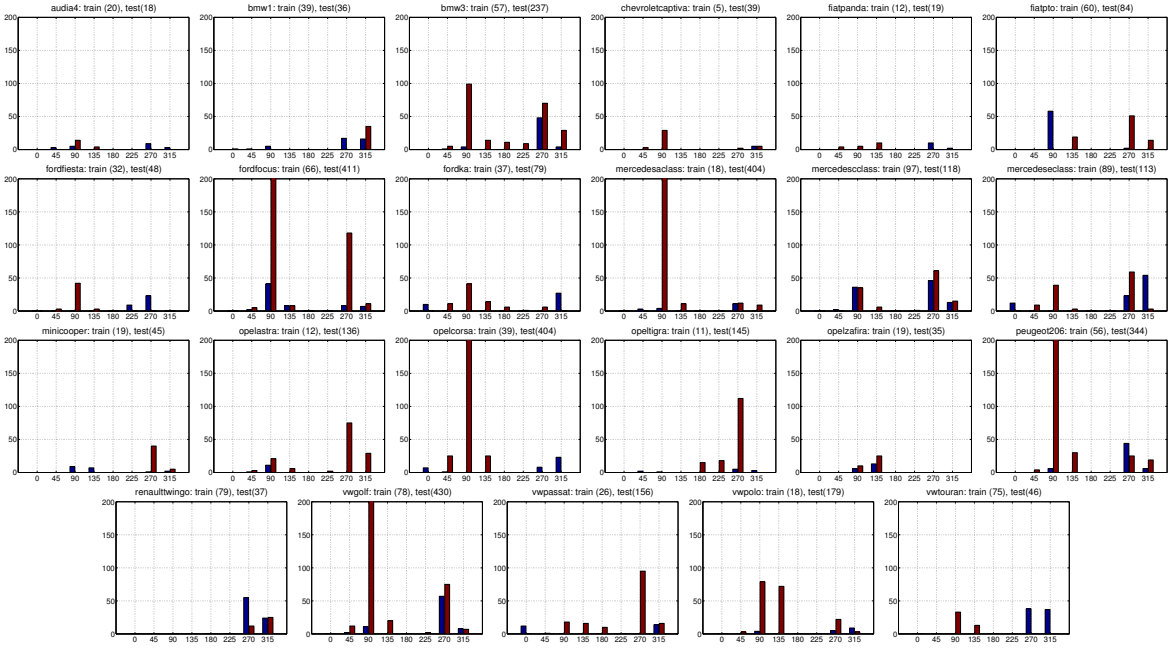


Figure 8.6: *Car-models* train (blue) and test (red) statistics over 8 viewpoint bins.

8.4 CONCLUSION

In this chapter, we focused on fine-grained and multi-view object representations with the aim of improving general object class detection. Specifically, we approached the problem of scarce and unbalanced training data for training multi-view and fine-grained detectors from a transfer learning perspective, introducing two flavors of learning prior distributions over permissible detectors, one based on sparse feature correlations, and one based on the full covariance matrix between all features. In both cases, we have demonstrated improved simultaneous 2D bounding box localization and viewpoint estimation performance when applying these priors to detectors based on basic-level category representations. In addition, the second flavor allowed us to learn reliable detectors even for finer-grained object class representations, resulting in an additional boost in performance on a realistic dataset of street scenes (Geiger *et al.*, 2012).

9 OCCLUSION PATTERNS FOR OBJECT CLASS DETECTION

Contents

Occlusions are one of the main source of errors in many vision applications. As in this thesis, richer object representations set the path towards high-level applications like 3D scene understanding, in this chapter we approach occlusion as a "first class citizen" and build explicit occlusion representations. While in previous chapters we explored 3D and fine-grained object representations, here we focus on occlusion representations and we leave the beaten path of methods that treat occlusion as just another source of noise – instead, we treat occlusions as signal that should be modeled, and explicitly represent them in an object class model. Specifically, we include the occluder itself in the object representation, by mining distinctive, reoccurring *occlusion patterns* from annotated training data. The patterns represent modes of the object appearance distributions that are subsequently modeled by several detectors with varying degree of sophistication. In particular, we evaluate and compare models that range from standard object class detectors to hierarchical, part-based representations of occluder/occludee pairs. We experimentally confirm the benefits of our explicit occlusion model in terms of increased occluded objects detection, but also in terms of overall detection performance. The resulting object detector was the winner of the object class detection benchmark of the Reconstruction Meets Recognition Challenge (RMRC) in 2013.

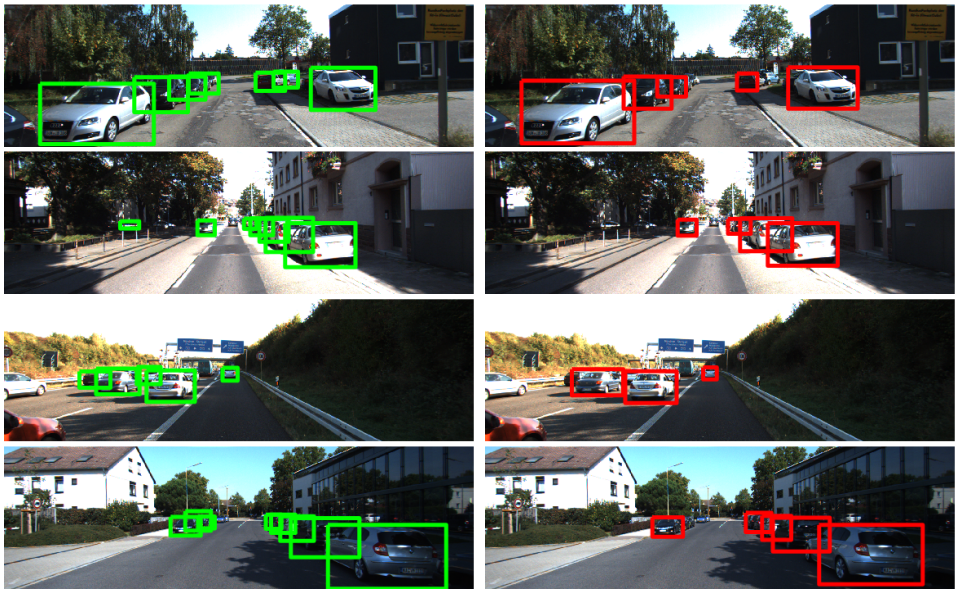


Figure 9.1: Detections on the KITTI dataset (Geiger *et al.*, 2012). (Left) True positive detections by our occluded objects detector. Even hard occlusion cases are detected. (Right) True positives by the DPM (Felzenszwalb *et al.*, 2010).

9.1 INTRODUCTION

Object class recognition has made remarkable progress in recent years (Everingham *et al.*, 2010), both on the level of individual classes (Dalal and Triggs, 2005; Felzenszwalb *et al.*, 2010) and on the level of entire visual scenes (Wojek *et al.*, 2011; Bao and Savarese, 2011). Reminiscent of the early days of computer vision, 2D bounding box-based localization has been generalized to more fine-grained object class representations capable of predicting poses (Andriluka *et al.*, 2009; Yang *et al.*, 2012), viewpoints (Savarese and Fei-Fei, 2007), 3D parts (3D^{2}PM), and fine-grained categories (Stark *et al.*, 2012).

Despite these achievements towards more accurate object hypotheses, *partial occlusion* still poses a major challenge to state-of-the-art detectors (Dalal and Triggs, 2005; Felzenszwalb *et al.*, 2010), as becomes apparent when analyzing the results of current benchmark datasets (Everingham *et al.*, 2010). While there have been attempts to tackle the occlusion problem by integrating detection with segmentation (Gao *et al.*, 2011) and latent variables for predicting truncation (Vedaldi and Zisserman, 2009; Wang *et al.*, 2009) resulting in improved recognition performance, all these attempts have been tailored to specific kinds of detection models, and not been widely adopted by the community.

Curiously, what is also common to these approaches is that they focus entirely on the occluded object – *the occludee* – without an explicit notion of the cause of occlusion. While this is more general than assuming a specific type of occlusion, it also complicates the distinction between weak, but visible evidence for an object

and an occluder. Here we therefore follow a different route, by treating the *occluder* as a first class citizen in the occlusion problem. In particular, we start from the observation that certain types of occlusions are more likely than others: consider a street scene with cars parked on either side of the road (as in Figure 9.1). Clearly, the visible and occluded portions of cars tend to form patterns that repeat numerous times, providing valuable visual cues about both the presence of individual objects and the layout of the scene as a whole.

Based on this observation, we chose to explicitly model these *occlusion patterns* by leveraging fine-grained, 3D annotations of a recent data set of urban street scenes (Geiger *et al.*, 2012). In particular, we mine reoccurring spatial arrangements of objects observed from a specific viewpoint, and model their distinctive appearance by an array of specialized detectors. To that end, we evaluate and compare two different models: i) a single-object class detector specifically trained to detect occluded objects from multiple viewpoints, occluded by various occluders, ii) a hierarchical double-object detector explicitly trained for accurate occluder/occludee bounding box localization. As baselines we include a standard, state-of-the-art object class detector (Felzenszwalb *et al.*, 2010) as well as a recently proposed double-person detector (Tang *et al.*, 2012) in the evaluation, with sometimes surprising results (Section 9.4).

This chapter makes the following contributions. First, we approach the challenging problem of partial occlusions in object class recognition from a different angle than most recent attempts by treating causes of occlusions as first class citizens in the model. Second, we propose three different implementations of this notion of varying complexity, ranging from easily implementable out-of-the-box solutions to powerful, hierarchical models of occluder/occludee pairs. And third, in an extensive experimental study we evaluate and compare these different techniques, providing insights that we believe to be helpful in tackling the partial occlusion challenge in a principled manner.

9.2 OCCLUSION PATTERNS

Our approach to modelling partial occlusions is based on the notion of *occlusion patterns*, i.e., re-occurring arrangements of objects that occlude each other in specific ways and that are observed from a specific viewpoint. Note that a similar approach has been taken in the poselet framework (Bourdev and Malik, 2009), but in the context of human body pose estimation and the resulting problem of dealing with self-occlusion.

Specifically, we limit ourselves to pairs of objects, giving rise to occlusion patterns on the level of single objects (*occludees*) and double objects (*occluder-occludee* pairs).

Figure 9.2: Visualization of mined *occlusion patterns* (occluder-occludee pairs). Top to bottom: 3D bounding box annotations provided by KITTI (Geiger *et al.*, 2012) for the cluster centroid along with the objects azimuth (row (1)), the corresponding average image over all cluster members (row (2)), two cluster members with corresponding 2D bounding boxes of occluder, occludee, and their union (rows (3) - (4)). Occlusion patterns span a wide range of occluder-occludee arrangements: resulting appearance can be well aligned (leftmost columns), or diverging (rightmost columns) – note that occluders are sometimes themselves occluded.

9.2.1 Mining occlusion patterns

We mine occlusion patterns from training data by leveraging fine-grained annotations in the form of 3D object bounding boxes and camera projection matrices that are readily available as part of the KITTI dataset (Geiger *et al.*, 2012). We use these annotations to define a joint feature space that represents both the relative layout of two objects taking part in an occlusion and the viewpoint from which this arrangement is observed by the camera. We then perform clustering on this joint feature space, resulting in an assignment of object pairs to clusters that we use as training data for the components of mixture models, as detailed in Sec. 9.3.

Feature representation. We use the following properties of occlusion patterns as features in our clustering: i) occluder left/right of occludee in image space, ii) occluder and occludee orientation in 3D object coordinates, iii) occluder is/is not itself occluded, iv) degree of occlusion of occludee.

Rule-based clustering. We found that a simple, greedy clustering scheme based on repeatedly splitting the training data according to fixed rules (e.g. based on assigning the viewing angle of the occluder to one of a fixed number of predetermined bins) resulted in sufficiently clean clusters. Figure 9.2 visualizes a selection of occlusion patterns mined from the KITTI dataset. As shown by the average images over cluster members (row (2)), some occlusion patterns are quite well aligned, which is a prerequisite for learning reliable detectors from them (Sec. 9.4.2).

9.3 OCCLUSION PATTERN DETECTORS

In the following, we introduce three different models for the detection of occlusion patterns, each based on the well known and tested deformable part model (DPM (Felzenszwalb *et al.*, 2010)) framework. We propose two qualitatively different types of models. The first type (Section 9.3.2) focuses on *individual* occluded objects, by dedicating distinct mixture components to different single-object occlusion patterns. The second type (Section 9.3.3) models *pairs* of objects in occlusion interaction, i.e. modelling both occluder and occludee. For the second model we propose two different variants (a symmetric and an a-symmetric one).

9.3.1 Preliminaries

We briefly recap the basics of the DPM model as implemented in (Felzenszwalb *et al.*, 2010). The DPM is a mixture of C star shaped log-linear conditional random fields (CRF), all of which have a root p_0 and a number of latent parts $p_i, i = 1, \ldots, M$. All parts are parameterized through their *l*eft, *r*ight, *t*op and *b*ottom extent (l, r, t, b). This defines both position and aspect ratio of the bounding box. Root and latent parts are singly connected through pairwise factors. The energy of a part configuration $p = (p_0, \ldots, p_M)$ given image evidence I for mixture component c is then

$$E_c(p; I) = \sum_{i=0}^{M} \langle v_i^c, \phi(p_i; I) \rangle + \sum_{i=1}^{M} \langle w_i^c, \phi(p_0, p_i) \rangle. \tag{9.1}$$

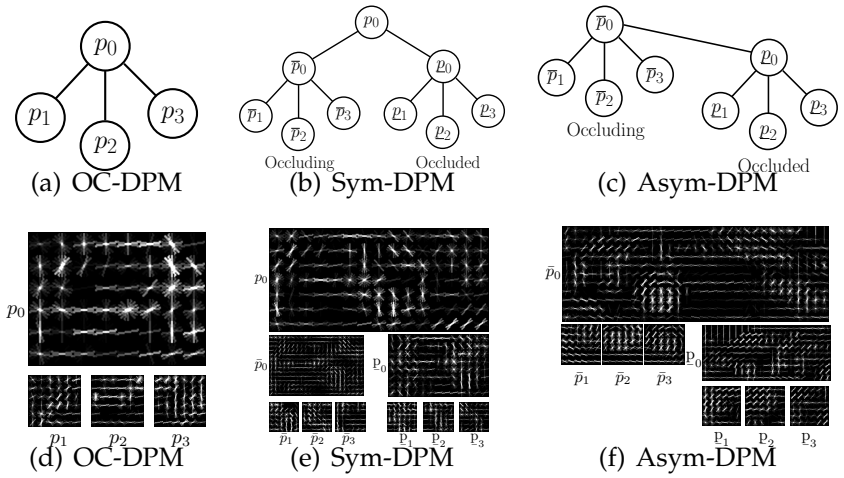
Each component has its own set of parameters (v^c, w^c) for unary and pairwise factors. The collection of those $c = 1, \ldots, C$ define the set of parameters that are learned during training. Training data is given as a set of N tuples $(I_n, y_n), n = 1, \ldots, N$ of pairs of images I and object annotations y, consisting of bounding boxes (l_n, r_n, t_n, b_n) and coarse viewpoint estimates.

9.3.2 Single-object occlusion patterns – OC-DPM

We experiment with the following extension of the DPM (Felzenszwalb *et al.*, 2010). In addition to the original components $c = 1, \ldots, C_{visible}$ that represent the appearances of instances of an object class of interest, we introduce additional mixture components dedicated to representing the distinctive appearance of *occluded* instances of that class. In particular, we reserve a distinct mixture components, for each of the *occludee* members of clusters resulting from our occlusion pattern mining step (Sec. 9.2).

9.3.3 Double-object occlusion patterns

While the single-object occlusion model of Sec. 9.3.2 has the potential to represent distinctive occlusion patterns in the data, modelling occluder and corresponding



(a) OC-DPM (b) Sym-DPM (c) Asym-DPM

(d) OC-DPM (e) Sym-DPM (f) Asym-DPM

Figure 9.3: Visualization of a single component of the three different occlusion models (a) OC-DPM, (b) the Sym-DPM (c) Asym-DPM as Sym-DPM but without a joint root variable. All models are shown with only three latent parts to avoid overloading the figure. The bottom row (d),(e),(f) show the learnt filters for the respective models. Note that for the Sym-DPM we place the joint root p_0 at half the resolution in the pyramid.

occludee *jointly* suggests a potential improvement: intuitively, the strong evidence of the occluder should provide strong cues as to where to look for the occludee. In the following we capture this intuition by designing two variants of a hierarchical occlusion model based on the DPM (Felzenszwalb *et al.*, 2010) framework. In these models occluder and occludee are allowed to move w.r.t. a spatial models much like parts in the DPM (Felzenszwalb *et al.*, 2010). The two models vary in their choice of topology of the associated spatial deformations. We note that a similar route has been explored by Tang *et al.* (2012), but in the context of people tracking.

9.3.3.1 *Double-objects with joint root – Sym-DPM*

The first double-object occlusion pattern detector is graphically depicted in Figure 9.3 (b,e). The idea is to join two star shaped CRFs, one for the *occluding* object $\overline{p}_0$, and one for the *occluded* object $\underline{p}_0$ by an extra common root part $p_0 = (l, r, t, b)$. As training annotation for the root part we use the tightest rectangle around the union of the two objects, see the green bounding boxes in Figure 9.2. The inclusion of this common root part introduces three new terms to the energy, an appearance term for the common root $\langle v^c_{joint}, \phi(p_0; I)\rangle$ and two pairwise deformation terms

$$\langle \underline{w}, \phi(\underline{p}_0, p_{joint})\rangle + \langle \overline{w}, \phi(\overline{p}_0, p_{joint})\rangle \tag{9.2}$$

with new parameters $\underline{w}, \overline{w}$. For these pairwise terms we use the same feature function ϕ as for all other root-latent part relations in the DPM, basically a Gaussian factor on the displacement around an anchor point.

This model retains the properties of being singly connected and thus warrants tractable exact inference. Because of the form of the pairwise term one can still use the distance transform for efficient inference. We will refer to this model as Sym-DPM. During training we have annotations for three parts $\underline{p}_0, \overline{p}_0, p_0$, while all others remain latent.

9.3.3.2 *Double-objects without joint root – Asym-DPM*

The second double-object model is a variation of Sym-DPM, where the common root part is omitted (Figure 9.3 (c,f)). Instead, we directly link *occluder* and *occludee*. This relationship is asymmetric – which is why we refer to this model as Asym-DPM – and follows the intuition that the occluder can typically be trusted more (because it provides unhampered image evidence).

9.3.4 Training

All models that we introduced are trained using the structured SVM formulation as done for the DPM in Chapter 5. To avoid cluttering the notation we write the problem in the following general form

$$\begin{aligned} \min_{\beta,\xi\geq 0} \quad & \frac{1}{2}\|\beta\|^2 + \frac{C}{N}\sum_{n=1}^{N}\xi_n \\ \text{sb.t.} \quad & \max_h \langle \beta, \phi(I_n, y_n, h_n)\rangle - \max_{h'} \langle \beta, \phi(I_n, y', h')\rangle \\ & \ldots \geq \Delta(y_n, y') - \xi_n, \forall y' \in \mathcal{Y}. \end{aligned} \tag{9.3}$$

For the models considered here, β refers to all their parameters $(v, w, \underline{w}, \overline{w})$ for all components c, y to the bounding box annotations per example (can be 1 or 2), and h to the latent part placements. For simplicity we will use y as the bounding box annotation that could comprise one or two annotations/detections. This problem is a quadratic problem and can be solved using the CCCP algorithm, alternating between fixing the latent part assignments h', and updating the parameters. The latter step involves detecting high scoring bounding boxes and latent part assignments (y', h') using loss-augmented inference $(y', h') = \text{argmax}_{y,h}\langle \beta, \phi(I_n, y', h')\rangle + \Delta(y_n, y')$.

The most important change w.r.t. learning a DPM through SSVM compared to Chapter 5 is that the loss now has to take into account the possibility of multiple annotations and predictions. We use the standard intersection over union loss Δ_{VOC} for a pair of bounding boxes y, y'

$$\Delta_{VOC}(y, y') = (1 - \frac{y \cap y'}{y \cup y'}). \tag{9.4}$$

and modify it in the following way. There are four different cases that have to be distinguished, 1 or 2 objects in the annotation and 1 or 2 objects that are being predicted.

In case the model predicts a single bounding box y only (decided through the choice of the component) the loss is the intersection over union loss between $\Delta(y_n, y)$ in case there is one annotation and $\Delta(\overline{y}_n, y)$ in case of an occlusion annotation. This of course is not ideal, since in case there is a second occluded object that is not being predicted, this will result in a false negative detection.

When two bounding boxes are predicted $\overline{y}, \underline{y}$ the loss is computed as either $\Delta(y_n, \overline{y})$ in case there is a single annotation or as the average $0.5\Delta(\overline{y}_n, \overline{y}) + 0.5\Delta(\underline{y}_n, \underline{y})$ between occluding and occcluded object. Again this is a proxy only, since the case of two detections but only one present in the annotation would result in a false positive.

As explained, our loss is not a perfect match since it does not penalize all false positives/negatives. We still believe it is a better proxy than the Hinge loss and found while experimenting with different implementations of Δ that the choice of Δ has only a small influence on the test time performance. This is consistent with the findings of Chapter 5 who report that the "correct" structured loss function for the DPM that takes into account the bounding box prediction rather than using the Hinge loss for classification (Felzenszwalb *et al.*, 2010) gives a consistent but rather small improvement. Our implementation of the loss function is capturing both single and double object detections simultaneously.

Detection and non-maximum suppression Test time inference in all mentioned models is tractable and efficient because they still are singly connected and allow the use of the distance transform. As usual we compute the max-marginal scores for the root components, p_0, and $\underline{p}_0, \overline{p}_0$ resp. Non-maximum suppression is done in the standard way.

9.4 EXPERIMENTS

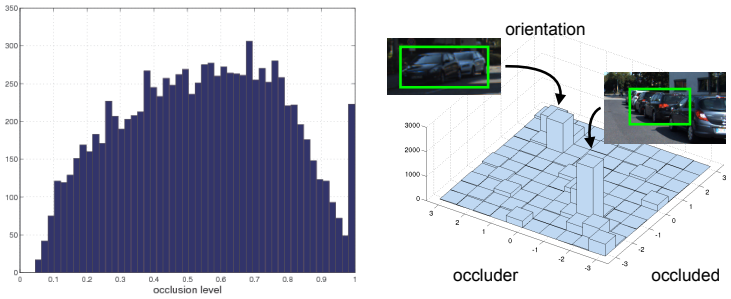
In the following, we give a detailed analysis of the various methods based on the notion of occlusion patterns that we introduced in Section 9.2. In a series of experiments we consider both results according to classical 2D bounding box-based localization measures, as well as a closer look at specific occlusion cases. We commence by confirming the ability of our models to detect occlusion patterns in isolation 9.4.2, and then move on the task of object class detection in an unconstrained setting, comprising both un-occluded and occluded objects of varying difficulty 9.4.3.

9.4.1 Data set

We chose the recently proposed KITTI data set (Geiger *et al.*, 2012) as the testbed for our evaluation, since it provides a large variety of challenging, real-world imagery of occlusion cases of different complexity, and comes with fine-grained annotations (manual 3D BBs of Lidar scans) that support a detailed analysis. It contains 7481 images of street scenes with accompanying Lidar scans, acquired from a moving vehicle. It is divided into 151 distinct sequences with varying duration. The sequences mostly depict inner-city scenes, but also contain rural and highway passages. In all

	#objects	#occluded objects	%
Car	28521	15231	53.4
Pedest.	4445	1805	40.6
Cycles	1612	772	44.5

Table 9.1: KITTI dataset statistics on objects and occlusions



(a) Occlusion histogram (b) Orientation histogram

Figure 9.4: Occlusion and orientation histograms

experiments we limit ourselves to a thorough evaluation of the *Car* object class (since it occurs most often), but give additional results on the *Pedestrian* class, highlighting that our approach generalizes to non-rigid objects.

Protocol. In all experiments we perform k-fold cross-validation on the publicly available data set portion in all experiments ($k = 3$). We successively train models on two folds, evaluate them on the other fold, and afterwards aggregate the per-fold results on the level of detections.

Occlusion statistics. The KITTI dataset (Geiger *et al.*, 2012) is a rich source of challenging occlusion cases, as shown in Tab. 9.1. It contains thousands of objects of which almost half are occluded, e.g. 53.4% of 28521 Car objects. From Figure 9.4 (a), we see that many of these are occluded to a substantial degree (the mode is around 60% occlusion). Further, Figure 9.4 (b) confirms our intuition that occlusions tend to form patterns: the distribution over relative orientations of occluder-occludee pairs of cars is highly peaked around two modes.

In all our experiments on *Car* (*Pedestrian*) we train our occlusion models with 6 (6) components for visible objects and 16 (15)[8] components for occlusion patterns. We obtain these numbers after keeping the occlusion pattern clusters which have at least 30 positive training examples.

9.4.2 Detecting occlusion patterns

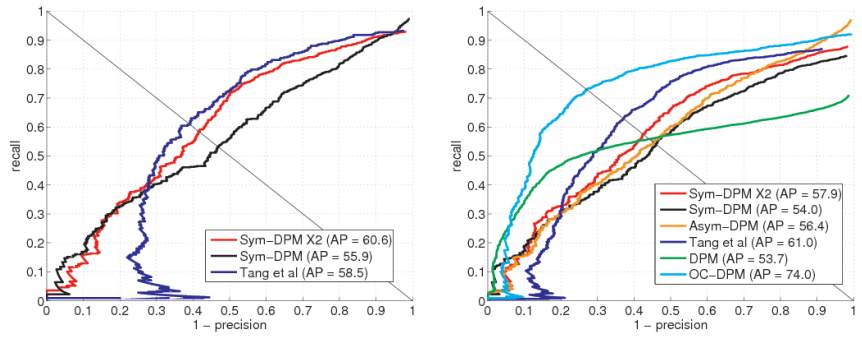
We commence by evaluating the ability of our models to reliably detect occlusion patterns in isolation, since this constitutes the basis for handling occlusion cases in a

[8]The numbers vary for different folds

realistic detection setting (Section 9.4.3). In particular, we contrast the performance of our models (OC-DPM, Sym-DPM, and Asym-DPM) with two baselines, the standard deformable part model (Felzenszwalb *et al.*, 2010), unaware of occlusions, and our implementation of the recently proposed double-person detector (Tang *et al.*, 2012), which we adapt to the *Car* setting.

Double-object occlusion patterns. We first consider the joint detection of occlusion patterns in the form of object pairs (occluder and occludee). For that purpose, we limit our evaluation to a corresponding subset of the test data, i.e. images that contain occlusion pairs, which we determine from the available fine-grained annotations (we run the occlusion pattern mining of Section 9.2 with parameters that yield a single cluster). This targeted evaluation is essential in order to separate concerns, and to draw meaningful conclusions about the role of different variants of occlusion modelling from the results. Figure 9.5 (left) gives the corresponding results, comparing the performance of two variants of our Sym-DPM model (normal, in black, and a variant with object-level templates at doubled HOG resolution, red) to the double-person detector of (Tang *et al.*, 2012) (magenta). We make the following observations: first, we observe that all detectors achieve a relatively high recall of over 90% – note that this can not be trivially achieved by lower detection thresholds, since different occlusion patterns result in largely different aspect ratios, which our models counter by dedicating a discrete set of distinct components to them. Second, we observe that our Sym-DPM performs on a comparable level to the baseline Tang *et al.* (2012) (55.9% vs. 58.5% AP), and dominates in its double-resolution variant (60.6% AP).

Single-object occlusion patterns. Based on the setup of the previous experiment we turn to evaluating our occlusion pattern detectors on the level of individual objects (this comprises both occluders and occludees from the double-object occlusion patterns). To that end, we add our single-object detectors to the comparison, namely, our Asym-DPM (orange), our OC-DPM (cyan), and the deformable part model (Felzenszwalb *et al.*, 2010) baseline (green). Figure 9.5 (right) gives the corresponding results. Clearly, all explicit means of modelling occlusion improve over the



(a) Double object detection (b) Single object detection

Figure 9.5: (a) Joint , (b) single *Car* detection results

DPM (Felzenszwalb *et al.*, 2010) baseline (53.7% AP) by up to a striking 20.3% AP (OC-DPM, cyan, 74% AP). Equally, the recall improves drastically from approx. 70% to over 80%. As concerns the relative performance of the different occlusion models, we observe a different ordering compared to the double-object occlusion pattern case: the double-object baseline Tang *et al.* (blue, 61% AP) performs slightly better than our double-resolution Sym-DPM (red, 57.9% AP), followed by our Asym-DPM (orange, 56.4% AP), and our normal Sym-DPM (black, 54.0 AP). Curiously, the arguably simplest model, our OC-DPM, outperforms all other models by at least to 13% AP.

Summary. To summarize, we conclude that detecting occlusion patterns in images is in fact feasible, achieving both sufficiently high recall (over 90% for both single- and double-object occlusion patterns) and reasonable AP (up to 74% for single-object occlusion patterns). We consider this result viable evidence that occlusion pattern detectors have the potential to aid recognition in the case of occlusion (which we examine and verify in Section 9.4.3). Furthermore, careful and explicit modelling of occluder and occludee characteristics helps for the joint detection of double-object patterns (our hierarchical Sym-DPM model outperforms the flat baseline Tang *et al.*). For the single-object case, however, the simplest model OC-DPM outperforms all others by a significant margin.

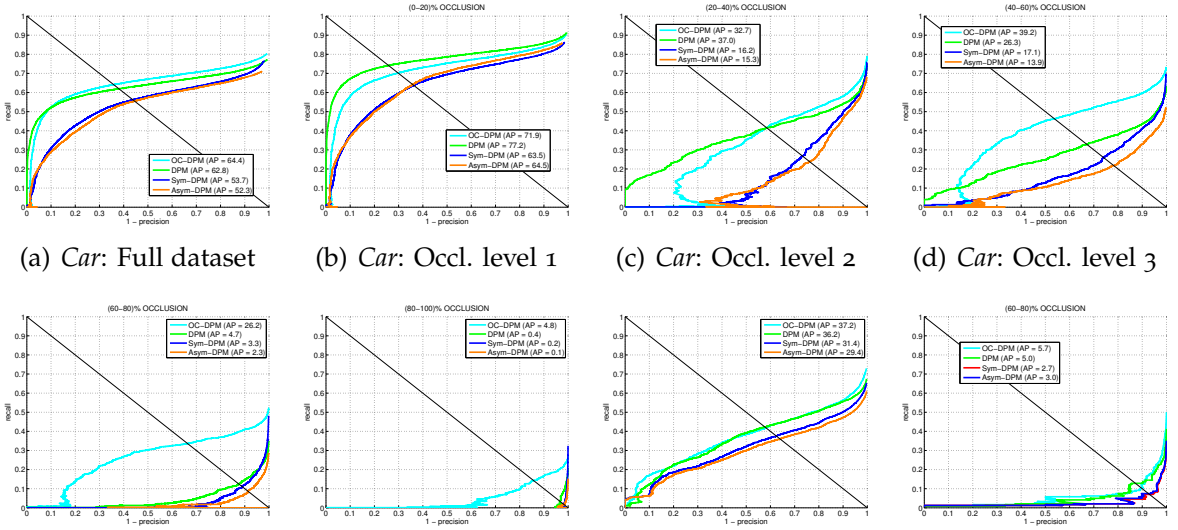
9.4.3 Occlusion patterns for object class detection

In this section we apply our findings from the isolated evaluation of occlusion pattern detectors to the more realistic setting of unconstrained object class detection, again considering the KITTI dataset (Geiger *et al.*, 2012) as a testbed. Since the focus is again on occlusion, we consider a series of increasingly difficult scenarios for comparing performance, corresponding to increasing levels of occlusion (which we measure based on 3D annotations and the given camera parameters). Specifically, we consider the following six scenarios: the full, unconstrained data set (Figure 9.6 (a)), the data set restricted to at most 20% occluded objects (Figure 9.6 (b)), restricted to objects occluded between 20 and 40% (Figure 9.6 (c)), between 40 and 60% (Figure 9.6 (d)), between 60 and 80% (Figure 9.6 (e)), and beyond 80% (Figure 9.6 (f)).

Modeling unoccluded objects. In order to enable detection of occluded as well as unoccluded object instances, we augment our various occlusion pattern detectors by additional mixture components for unoccluded objects.

Results - full dataset. On the full data set (Figure 9.6 (a)) we observe that the trends from the isolated evaluation of occlusion patterns (Section 9.4.2) transfer to the more realistic object class detection setting: while the double-object occlusion pattern detectors are comparable in terms of AP (Asym-DPM, orange, 52.3%; Sym-DPM, blue, 53.7%), our OC-DPM achieves the best performance (64.4%), improving over the next best double-object occlusion pattern detector Sym-DPM by a significant margin of 10.7%.

Surprisingly, the DPM (Felzenszwalb *et al.*, 2010) baseline (green, 62.8% AP) beats



(a) *Car*: Full dataset (b) *Car*: Occl. level 1 (c) *Car*: Occl. level 2 (d) *Car*: Occl. level 3

(e) *Car*: Occl. level 4 (f) *Car*: Occl. level 5 (g) *Pedestrian*: Full dataset (h) *Pedestrian*: Occl. level 4

Figure 9.6: Detection performance for class *Car* on (a) the full dataset, (b)-(f) increasing occlusion levels from $[0-20]\%$ to $[80-100]\%$. Detection performance on class *Pedestrian*, (g) full set, (h) $[60-80]\%$ occlusion.

all double-object occlusion pattern detectors, but is in turn outperformed by our OC-DPM (cyan, 64.4%). While the latter improvement seems modest at first glance, we point out that this corresponds to obtaining 1000 more true positive detections, which is approximately the number of cars (1250) in the entire Pascal VOC 2007 trainval set.

In comparison to Tang *et al.* (53.9%), Sym-DPM and Asym-DPM provide similar performance. All double-object detectors have proven to be very sensitive to the non-maxima suppression scheme used and suffer from score incomparability among the double and single object components. We intend to address this issue in future work.

On the *Pedestrian* class (Figure 9.6 (g)) OC-DPM (37.2%) outperform the DPM (36.2%), confirming the benefit of our occlusion modelling, while Sym-DPM (31.4%) outperforms the Asym-DPM (29.4%).

Results - occlusion. We proceed by examining the results for increasing levels of occlusion (Figure 9.6 (b-f)), making the following observations. First, we observe that the relative ordering among double-object and single-object occlusion pattern detectors is stable across occlusion levels: our OC-DPM (cyan) outperforms all double-object occlusion pattern detectors, namely, Sym-DPM (blue) and Asym-DPM (orange). Second, the DPM (Felzenszwalb *et al.*, 2010) baseline (green) excels at low levels of occlusion (77.2% AP for up to 20% occlusion, 37% AP for 20 to 40% occlusion), performing better than the double-object occlusion pattern detectors for all occlusion levels. But third, the DPM (Felzenszwalb *et al.*, 2010) is outperformed by our OC-DPM for all occlusion levels above 40% by significant margins (12.9%,

AP	Easy	Moderate	Hard
OC-DPM	**74.9**	**66.0**	**53.9**
LSVM-MDPM-sv	68.0	56.5	44.2
LSVM-MDPM-us	66.5	55.4	41.0
mBoW	36.0	23.8	18.4

Table 9.2: KITTI testing set results (Geiger *et al.*, 2012). *Car* category.

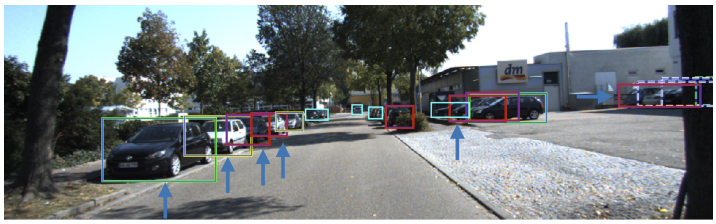


Figure 9.7: Examples of non tight BB annotations

21.5%, and 4.4% AP, respectively).

The same trend can be observed for the *Pedestrian* class: for occlusions between 60 and 80% OC-DPM (5.7%) outperforms DPM (5.0%) (Figure 9.6 (h)). Asym-DPM (3.0%) outperforms the Sym-DPM (2.7%).

Summary. We conclude that occlusion pattern detectors can in fact aid detection in presence of occlusion, and the benefit increases with increasing occlusion level. While, to our surprise, we found that double-object occlusion pattern detectors were not competitive, our simpler, single-object occlusion pattern detector (OC-DPM) improved performance for occlusion by a significant margin.

9.4.4 KITTI testing results

Tab. 9.2 illustrates the OC-DPM *Car* detection performance on the KITTI testing set, compared to three baselines: LSVM-MDPM-sv, LSVM-MDPM-us (Geiger *et al.*, 2012) and mBoW (Behley *et al.*, 2013). The results confirm the superior OC-DPM performance, especially on the *Hard* data case (see Geiger *et al.* (2012) for details), where OC-DPM with 53.9% AP outperforms the LSVM-MDPM-sv. With this result we entered the Reconstruction Meets Recognition Challenge (RMRC) in 2013 and won the 1st place in the object detection task.

9.4.5 Discussion

In the course of our evaluation, we have gained a number of insights which we discuss in the following.

Biased occlusion statistics. From our experience, the poor performance of double-object occlusion detectors on the KITTI dataset (Geiger *et al.*, 2012) (Section 9.4.3), which is in contrast to Tang *et al.* (2012) findings for people detection, can

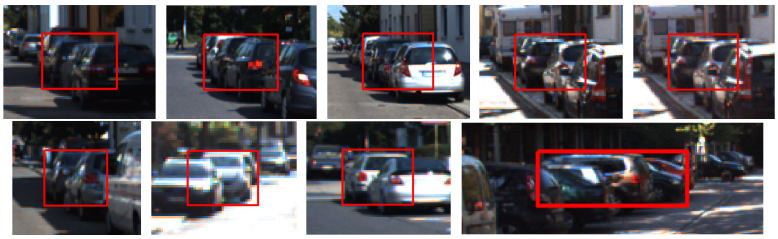


Figure 9.8: Valid detections on unannotated objects

be explained by the distribution over occlusion patterns: it seems biased towards extremely challenging "occluded occluder" cases. We found a large fraction of examples in which double-objects appear in arrangements of a larger number of objects (e.g. row of parked cars), where the occluder is itself occluded – these cases are not correctly represented by occluder-occludee models. In these cases it proves less robust to combine possibly conflicting pairwise detections (Asym-DPM, Sym-DPM) into a consistent interpretation than aggregating single-object occlusion patterns (OC-DPM). As a result, single-object models ((Felzenszwalb *et al.*, 2010), OC-DPM) tend to be more robust against this bias, resulting in improved performance.

Annotation noise. We also found that the KITTI dataset (Geiger *et al.*, 2012) contains a significant number of occluded objects that are not annotated, supposingly due to being in the Lidar shadow, and hence missing 3D ground truth evidence for annotation. While there is a reserved "don't care" region label for these cases, this seldomly overlaps sufficiently with the object bounding box in question. This is particularly true for our best performing OC-DPM model, for which the first ≈ 70 false positive detections are of that nature, resulting in a severe under-estimation of its performance in Section 9.4.3 (Figure9.8 shows examples).

Overlap criterion. In line with the previous argument we believe the overlap threshold of 70% intersection-over-union (Everingham *et al.*, 2010) proposed by the KITTI dataset (Geiger *et al.*, 2012) is hardly compatible with the accuracy of the annotations in many cases (Figure 9.7 gives examples), which is why we report results for the less challenging but more robust overlap of 50%.

9.5 CONCLUSION

In this chapter, we have considered the long-standing problem of partial occlusion by making *occluders* first class citizens. In particular, we have proposed two different models for detecting distinctive, reoccurring occlusion patterns, mined from annotated training data. Using these detectors we could improve over the performance of a current, state-of-the-art object class detector over an entire dataset of challenging urban street scenes, but even more so for increasingly difficult cases in terms of occlusion. Our most important findings are: i) reoccurring occlusion patterns can be automatically mined and reliably detected, ii) they can aid object detection, and iii) occlusion is still challenging also in terms of dataset annotation.

10 WHAT IS HOLDING BACK CONVNETS FOR DETECTION?

Contents

Convolutional neural networks are state-of-the-art computer vision technology today. Praised for the end-to-end representation learning, circumventing the need for manual engineering of representations, convnets have been considered to implicitly and gradually build invariances w.r.t. to various appearance factors. For example, a convnet trained for object recognition is deemed to build invariant representations to appearance factors like viewpoints (Lenc and Vedaldi, 2015), while at the same time it encodes object detectors (Zhou *et al.*, 2015) and object parts (Simon *et al.*, 2014) internally. Inspired by these intuitions, in this chapter we refrain from building richer object representations and instead, we dive into the direction of understanding convnet representations w.r.t. various appearance factors like viewpoints, shapes, size etc. Furthermore, we focus on understanding what is stopping convnets to further improve it's performance. Driven by the intuition that "bigger models and more data" is the way to improve convnets, we explore what have current state-of-the-art convnet architectures learned, and what are their weaknesses. In addition, we explore what could current architectures learn by generating additional, synthetic training data by rendering CAD models with varying degree of realism. In the end we combine the best practices, resulting in state-of-the-art performance on Pascal3D+ dataset.

10.1 INTRODUCTION

In the last years convolutional neural networks (convnets) have become "the hammer that pounds many nails" of computer vision. Classical problems such as general image classification (Krizhevsky *et al.*, 2012), object detection (Girshick *et al.*, 2014), pose estimation (Chen and Yuille, 2014), face recognition (Schroff *et al.*, 2015), object tracking (Li *et al.*, 2014), keypoint matching (Fischer *et al.*, 2014), stereo matching (Zbontar and LeCun, 2015), optical flow (Fischer *et al.*, 2015), boundary estimation (Xie and Tu, 2015), and semantic labelling (Long *et al.*, 2015), have now all top performing results based on a direct usage of convnets. The price to pay for such versatility and good results is a limited understanding of why convnets work so well, and how to build & train them to reach better results.

In this chapter we focus on convnets for object detection. For many object categories convnets have almost doubled over previous detection quality. Yet, it is unclear what exactly enables such good performance, and critically, how to further improve it. The usual word of wisdom for better detection with convnets is "larger networks and more data". But: how should the network grow; which kind of additional data will be most helpful; what follows after fine-tuning an ImageNet pre-trained model on the classes of interest? We aim at addressing such questions in the context of the R-CNN detection pipeline (Girshick *et al.*, 2014) (section 10.2).

Previous work aiming to analyse convnets have either focused on theoretical aspects (Bengio and Delalleau, 2011), visualising some specific patterns emerging inside the network (Le *et al.*, 2012; Simonyan *et al.*, 2014; Springenberg *et al.*, 2015; Mahendran and Vedaldi, 2015), or doing ablation studies of working systems (Girshick *et al.*, 2014; Chatfield *et al.*, 2014; Agrawal *et al.*, 2014). However, it remains unclear what is withholding the detection capabilities of convnets.

Contributions This chapter contributes a novel empirical exploration of R-CNNs for detection. We use the recently available Pascal3D+(Xiang *et al.*, 2014a) dataset, as well as rendered images to analyze R-CNNs capabilities at a more detailed level than previous work. In a new set of experiments we explore which appearance factors are well captured by a trained R-CNN, and which ones are not. We consider factors such as rotation (azimuth, elevation), size, category, and instance shape. We want to know which aspects can be improved by simply increasing the training data, and which ones require changing the network. We want to answer both "what did the network learn?" (section 10.5) and "what can the network learn?" (section 10.6 and section 10.7). Our results indicate that current convnets (AlexNet (Krizhevsky *et al.*, 2012), GoogleNet (Szegedy *et al.*, 2014a), VGG16 (Simonyan and Zisserman, 2015)) struggle to model small objects, truncation, and occlusion and are not invariant to these factors. Simply increasing the training data does not solve these issues. On the other hand, properly designed synthetic training data can help pushing forward the overall detection performance.

10.2 THE R-CNN DETECTOR

The remarkable convnet results in the ImageNet 2012 classification competition (Krizhevsky *et al.*, 2012) ignited a new wave of neural networks for computer vision. R-CNN (Girshick *et al.*, 2014) adapts such convnets for the task of object detection, and has become the de-facto architecture for state-of-the-art object detection (with top results on Pascal VOC (Everingham *et al.*, 2007) and ImageNet (Deng *et al.*, 2009)) and is thus the focus of attention in this chapter. The R-CNN detector is a three stage pipeline: object proposal generation (Uijlings *et al.*, 2013), convnet feature extraction, and one-vs-all SVM classification with bounding box regression. We refer to the original paper for details of the training procedure (Girshick *et al.*, 2014). Different networks can be used for feature extraction (AlexNet (Krizhevsky *et al.*, 2012), VGG (Chatfield *et al.*, 2014), GoogleNet (Szegedy *et al.*, 2014a)), all pre-trained on ImageNet and fine-tuned for detection. The larger the network, the better the performance. The SVM gains a couple of final mAP points compared to logistic regression used during fine-tuning (and larger networks benefit less from it (Girshick, 2015)).

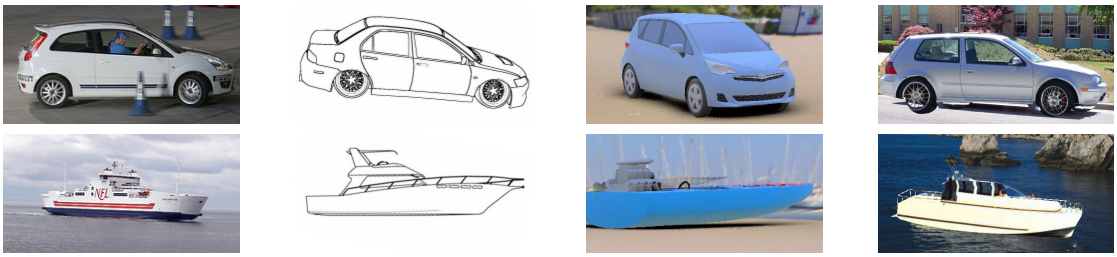
In this chapter we primarily focus on the core ingredient: convnet fine-tuning for object detection. We consider fine-tuning with various training distributions, and analyse the performance under various appearance factors. Unless otherwise specified reported numbers do not include the bounding box regression.

10.3 PASCAL3D+ DATASET

Our experiments are enabled by the recently introduced Pascal3D+ (Xiang *et al.*, 2014a) dataset. It enriches PASCAL VOC 2012 with 3D annotations in the form of aligned 3D CAD models for 11 classes (*aeroplane*, *bicycle*, *boat*, *bus*, *car*, *chair*, *diningtable*, *motorbike*, *sofa*, *train*, and *tv monitor*) of the *train* and *val* subsets. The alignments are obtained through human supervision, by first selecting the visually most similar CAD model for each instance, and specifying the correspondences between a set of 3D CAD model keypoints and their image projections, which are used to compute the 3D pose of the instance in the image. The rich object annotations include object pose and shape, and we use them as a test bed for our analysis. Unless otherwise stated all presented models are trained on the Pascal3D+ *train* set and evaluated on its test set (Pascal VOC 2012 *val*).

10.4 SYNTHETIC IMAGES

Convnets reach high classification/detection quality by using a large parametric model (e.g. in the order of 10^7 parameters). The price to pay is that convnets need a large training set to reach top performance. We want to explore whether the performance scales as we increase the amount of training data. To that end, we explore two possible directions to increase the data volume: data augmentation and



(a) Real image (b) Wire-frame (c) Plain texture (d) Text. transfer

Figure 10.1: Example training samples for different type of synthetic rendering.

synthetic data generation.

Data augmentation consists of creating new training samples by simple transformations of the original ones (such as scaling, cropping, blurring, subtle colour shifts, etc.), and it is a common practice during training on large convnets (Krizhevsky *et al.*, 2012; Chatfield *et al.*, 2014). To generate synthetic images we rely on CAD models of the object classes of interest. Rendering synthetic data has the advantage that we can generate large amounts of training data in a controlled setup, allowing for arbitrary appearance factor distributions. For our synthetic data experiments we use an extended set of CAD models, and consider multiple types of renderings (subsection 10.4.1).

Extended Pascal3D+ CAD models Although the Pascal3D+ dataset (Xiang *et al.*, 2014a) comes with its own set of CAD models, this set is rather small and it comes without material information (only polygonal mesh). Thus the Pascal3D+ models alone are not sufficient for our analysis. We extend this set with models collected from internet resources. We use an initial set of $\sim$40 models per class. For each Pascal3D+ training sample we generate one synthetic version per model using a "plain texture" rendering (see next section) with the same camera-to-object pose. We select suitable CAD models by evaluating the R-CNN (trained on Pascal 2007 train set) on the rendered images, and we keep a model if it generates the highest scoring response (across CAD models) for at least one training sample. This procedure makes sure we only use CAD models that generate somewhat realistic images close to the original training data distribution, and makes it easy to prune unsuitable models. Out of $\sim$440 initial models, $\sim$275 models pass the selection process ($\sim$25 models per class).

10.4.1 Rendering types

A priori it is unclear which type of rendering will be most effective to build or augment a convnet training set. We consider multiple options using the same set of CAD models. Note that all rendering strategies exploit the Pascal3D+ data to generate training samples with a distribution similar to the real data (similar size and orientation of the objects). See Figure 10.1 for example renderings.

Wire-frame Using a white background, shape boundaries of a CAD model are rendered as black lines. This rendering reflects the shape (not the mesh) of the object, abstracting its texture or material properties and might help the detector to focus on the shape aspects of the object.

Plain texture A somewhat more photo-realistic rendering considers the material properties (but not the textures), so that shadows are present. We considered using a blank background, or an environment model to generate plausible backgrounds. We obtain slightly improved results using the plausible backgrounds, and thus only report these results. This rendering provides "toy car" type images, that can be considered as middle ground between "wire frame" and "texture transfer" rendering.

Texture transfer All datasets suffer from bias (Torralba and Efros, 2011), and it is hard to identify it by hand. Ideally, synthetic renderings should have the same bias as the real data, while injecting additional diversity. We aim at solving this by generating new training samples via texture transfer. For a given annotated object on the Pascal3D+ dataset, we have both the image it belongs to and an aligned 3D CAD model. We create a new training image by replacing the object with a new 3D CAD model, and by applying over it a texture coming from a different image. This approach allows to generate objects with slightly different shapes, and with different textures, while still adequately positioned in a realistic background context (for now, our texture transfer approach ignores occlusions). This type of rendering is close to photo-realistic, using real background context, while increasing the diversity by injecting new object shapes and textures.

As we will see in section 10.7, any of our renderings can be used to improve detection performance. Still the level of realism affects how much improvement is obtained.

10.5 WHAT DID THE NETWORK LEARN FROM REAL DATA?

In this section we analyze R-CNNs detection performance in an attempt to understand what have the models actually learned. We first explore models performance across different appearance factors (subsection 10.5.1), going beyond the usual per-class detection performance. Second, we dive deeper and aim at understanding what have the network layers actually learned (subsection 10.5.2).

10.5.1 Detection performance across appearance factors

To analyze the performance across appearance factors we split each factor into equi-spaced bins. We present a new evaluation protocol where for each bin only the data falling in it are actually considered in the evaluation and the rest are ignored. This allows to dissect the detection performance across different aspects

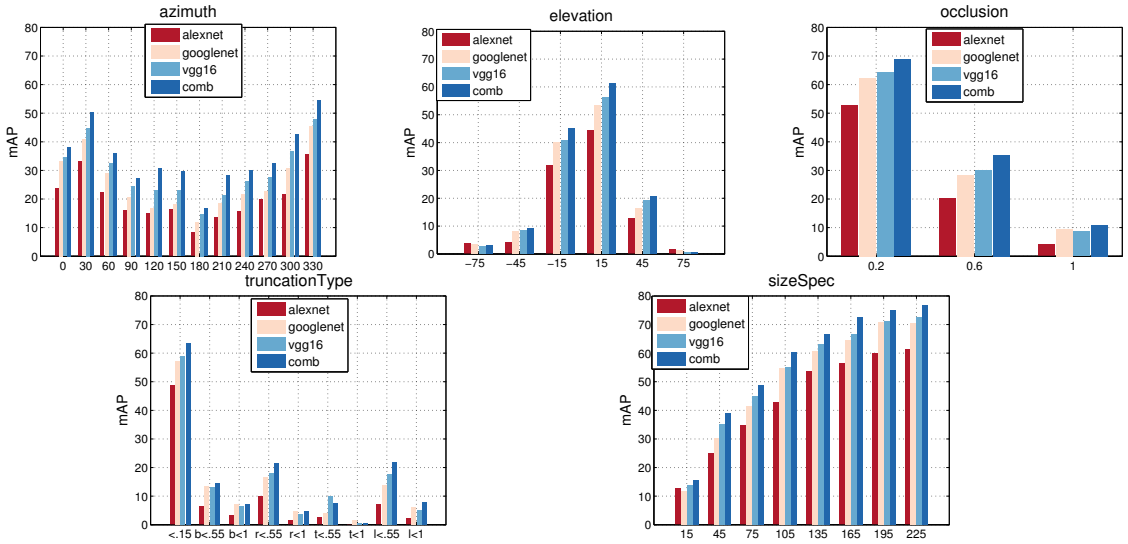
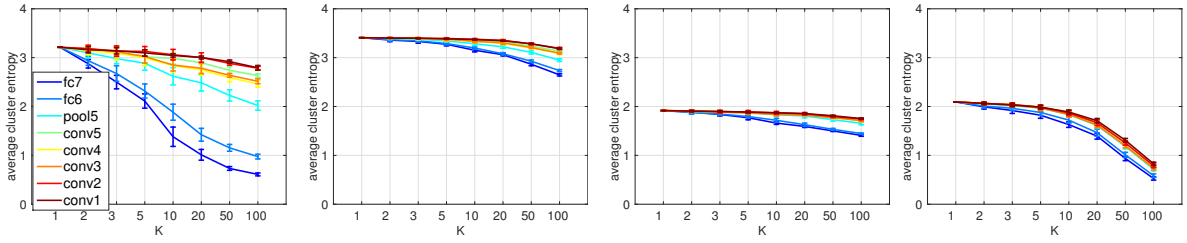


Figure 10.2: mAP of R-CNN over appearance factors. Pascal3D+.

of an appearance factor. The original R-CNN(Girshick *et al.*, 2014) work includes a similar analysis based on the toolkit from (Hoiem *et al.*, 2012). Pascal3D+ however enables a more fine-grained analysis. Our experiments report results for AlexNet (51.2 mAP)(Krizhevsky *et al.*, 2012), GoogleNet (56.6 mAP)(Szegedy *et al.*, 2014a), VGG16 (58.8 mAP)(Simonyan and Zisserman, 2015) and their combination (62.4 mAP).

Appearance factors We focus the evaluation on the following appearance factors: rotation (azimuth, elevation), size, occlusion and truncation as these factors have strong impact on objects appearance. Azimuth and elevation refer to the angular camera position w.r.t. the object. Size refers to the bounding box height. Although the Pascal3D+ dataset comes with binary occlusion and truncation states, using the aligned CAD models and segmentation masks we compute level of occlusion as well as level and type of truncation. While occlusion and truncation levels are expressed as object area percentage, we distinguish between 4 truncation types: bottom (b), top (t), left (l) and right (r) truncation.

Analysis Figure 10.2 reports performance across the factors. The results point to multiple general observations. First, there is a clear ordering among the models. VGG16 is better than GoogleNet on all factor bins, which in turn consistently outperforms AlexNet. The combination of the three models (SVM trained on concatenated features) consistently outperforms all of them suggesting there is underlying complementarity among the networks. Second, the relative strengths and weaknesses across the factors remain the same across models. All networks struggle with occlusions, truncations, and objects below 120 pixels in height. Third, for each factor the performance is not homogeneous across bins, suggesting the networks are not invariant w.r.t. the appearance factors.



(a) Class (b) Azimuth (c) Elevation (d) Shape

Figure 10.3: Average cluster entropy versus number of clusters K; at different layers, for different appearance factors. Pascal3D+ test data.

It should be noted that there are a few confounding factors in the results. First such factor is the image support (pixel area) of the object, which is strongly correlated with performance. Whenever the support is smaller e.g. small sizes, large occlusions/truncations or frontal views the performance is lower. Second confounding factor is the training data distribution. For a network with a finite number of parameters, it needs to decide to which cases it will allocate resources. The loss used during training will push the network to handle well the most common cases, and disregard the rare cases. Typical example is the elevation, where the models learn to handle well the near 0° cases (well represented), while they fail on the outliers: upper $(90^\circ)$and lower (-90°) cases. We explore this aspect in Section 10.6 by investigating performance under different training distributions.

Conclusion There is a clear performance ordering among the convnets which all have similar weaknesses, tightly related to data distribution and object area. Occlusion, truncation, and small objects are clearly weak points of the R-CNN detectors (arguably harder problems by themselves). Given similar tendencies next sections focus on AlexNet.

10.5.2 Appearance vector disentanglement

Other than just the raw detection quality, we are interested in understanding what did the network learn internally. While previous work focused on specific neuron activations (Goodfellow *et al.*, 2009), we aim at analyzing the feature representations of individual layers. Given a trained network, we apply it over positive test samples, and cluster the feature vectors at a given layer. We then inspect the cluster entropy with respect to different appearance factors, as we increase the number of clusters. The resulting curves are shown in Figure 10.3. Lower average entropy indicates that at the given layer the network is able to disentangle the considered appearance factor. Disentanglement relates to discriminative power, invariance, and equivariance. (Related entropy based metric is reported in Agrawal *et al.* (2014), however they focus on individual neurons).

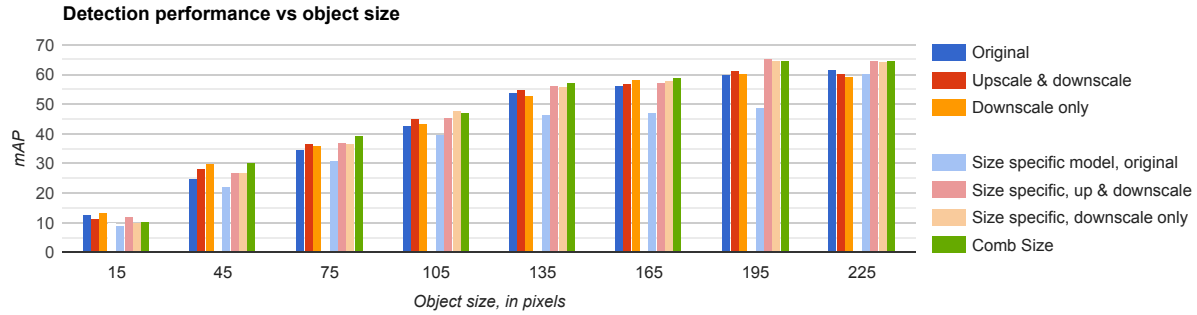


Figure 10.4: Training with varying object size distribution.

Analysis From Figure 10.3(a) we see that classes are well disentangled. As we go from the lowest conv1 layer to the highest fc7 layer the disentanglement increases, showing that with depth the network layers become more variant w.r.t. category. This is not surprising as the network has been trained to distinguish classes. On the other hand for azimuth, elevation and shape (class-specific disentanglement) the disentanglement across layers and across cluster number stays relatively constant, pointing out that the layers are not as variant to these factors.

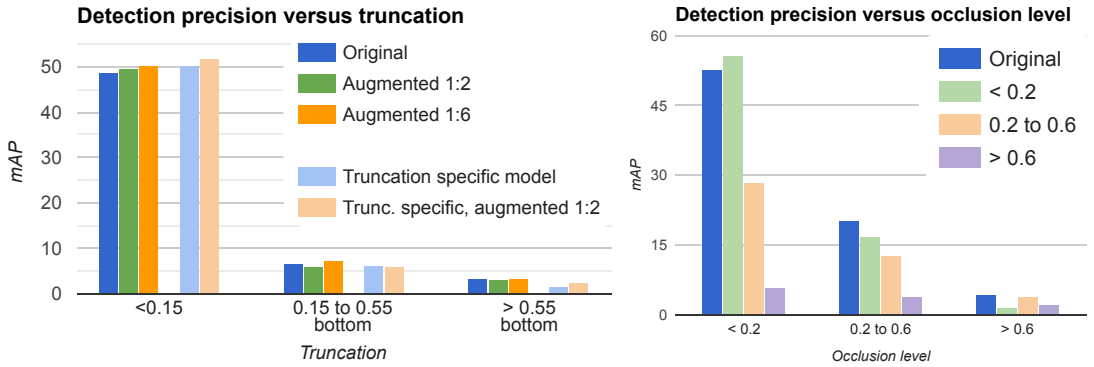
Conclusion We make two observations. First, convnet representations at higher layers disentangle object categories well, explaining its strong recognition performance. Second, network layers are to some extent invariant to different factors.

10.6 WHAT COULD THE NETWORK LEARN WITH MORE DATA?

Section 10.5 inspected what the network learned when trained with the original training set. In this section we explore what the network could learn if additional data is available. We will focus on size (subsection 10.6.1), truncations and occlusions (subsection 10.6.2) since these are aspects that R-CNNs struggle to handle. For each case we consider two general approaches: changing the training data distribution, or using additional supervision during training. For the former we use data augmentation to generate additional samples for specific size, occlusion, or truncation bins. Augmenting the training data distribution helps us realize if adding extra data for a specific factor bin helps improving the performance on that particular bin. When using additional supervision, we leverage the annotations to train a separate model for each bin. Providing an explicit signal during training forces the network to distinguish among specific factor bins. The experiments involve fine-tuning the R-CNN only as we are interested in convnet modelling capabilities.

10.6.1 Size handling

More data Figure 10.4 shows the results with different object size training distributions. The "original" bars correspond to the results in Figure 10.5.1. "Up &



(a) Truncation (b) Occlusion

Figure 10.5: Varying truncated and occluded training data distribution

Synthetic type	**Ratio** Real:Synth.	**mAP**
-	1:0	47.6
Wire-frame	0:1	21.8
Plain texture	0:1	23.5
Texture transfer	0:1	38.4
Wire-frame	1:2	48.3
Plain texture	1:2	49.9
Texture transfer	1:2	**51.5**

Table 10.1: Different synthetic data types

downscale" corresponds to training with a uniform size distribution across bins by up/down-scaling all samples to all bins. As upscaled images are blurry, "downscale only" avoids such blur, resulting in a distribution with more small size samples than larger sizes. Results in Figure 10.4 indicate that data augmentation can provide a few mAP points gain for small objects, however the network still struggles with small size, thus it is not invariant w.r.t. size despite the uniform training distribution.

Bin-specific models The right side bars of Figure 10.4 show results for bin-specific networks. Each bar corresponds to a model trained and tested on that size range. Both augmentation methods outperform the original data distribution on all size bins (e.g. at 195 pixels, "up & downscale" improves by 5.2 mAP). In "comb size" we combine the "up & downscale" size specific models via an SVM trained on their concatenated features. This results in superior overall performance (54.0 mAP) w.r.t. the original data (51.2 mAP with SVM).

Conclusion These results indicate that a) adding data uniformly across sizes provides mild gains for small objects and does not result in size invariant models, suggesting that the models suffer from limited capacity and b) training bin-specific

models results in better per bin and overall performance.

10.6.2 Truncation & occlusion handling

More data Figure 10.5(a) shows that generating truncated samples from non-truncated ones, respecting the original data distribution, help improve (1.5 mAP points) handling objects with minimal truncation; but does not improve medium or large truncation handling (trends for top, left and right are similar).

Bin-specific models Similar to the "more data" case, training a convnet for each truncation case only helps for the low truncation cases, but is ineffective for medium/large truncations. Similar to truncations, Figure 10.5(b) shows that specialising a network for each occlusion case is only effective for the low occlusions. Medium/high occlusions are a "distraction" for training non-occluded detectors.

Conclusion These results are a clear indication that training data do not help per-se handling these cases. Simply adding data or focusing the network on sub-tasks seems insufficient. Architectural changes to the detector seem required to obtain a meaningful improvement.

10.7 DOES SYNTHETIC DATA HELP?

We have seen that convnets have weak spots for object detection, and adding data results in limited gains. As convnets are data hungry methods, the question remains what happens when more data from the same distribution is introduced. Obtaining additional annotated data is expensive, thus we consider the option of using renderings. The results are summarised in Tab. 10.1. Again we focus on fine-tuning convnets only. All renderings are done using a similar data distribution as the original one, aiming to improve on common cases.

Analysis From Tab. 10.1 we observe that using synthetic data alone (0:1 ratio) under-performs compared to using real data, showing there is still room for improvement on the synthetic data itself. That being said, we observe that even the arguably weak wire-frame renderings do help improve detections when used as an extension of the real data. We empirically chose data ratio of 1:2 between real and synthetic as that seemed to strike good balance among the two data sources. As expected, the detection improvement is directly proportional to the photo-realism (see Tab. 10.1). This indicates that further gains can be expected as photo-realism is improved. Our texture transfer approach is quite effective, with a 4 mAP points improvement. Wire-frame renderings inject information from the extended CAD models. The plain texture renderings additionally inject material and background information. The texture transfer renderings use Pascal3D+ data, which include ImageNet images too. If we add these images directly to the training set (instead of

Data	CNN	mAP	AAVP
Pascal3D+	AlexNet	51.2	35.3
	GoogleNet	56.6	-
	VGG16	58.8	-
	comb	62.6	-
Pascal3D+ & Texture transfer	AlexNet	54.6	-
	GoogleNet	59.1	-
	VGG16	61.9	-
	comb	64.1	**43.8**
	comb+size	64.7	-
	comb+bb	66.3	-
	comb+size+bb	**67.2**	-

Table 10.2: Pascal3D+ results. Combining different convnetarchitectures.

doing texture transfer) we obtain 50.6 mAP (original to ImageNet images ratio is 1:3). This shows that the increased diversity of our synthetic samples further improve results. Plain textures provide 2 mAP points improvement, and texture transfer 4 mAP points. In comparison, Girshick (2015) reports 3 mAP points gain (on Pascal VOC 2012 test set) when using the Pascal VOC 2007 and the 2012 data. Our gains are quite comparable despite relying on synthetic renderings.

Conclusion Synthetic renderings are an effective mean to increase the overall detection quality. Even simple wire-frame renderings can be of help.

10.8 ALL-IN-ONE

In Tab. 10.2 we show results when training the SVM on top of the concatenated features of the convnets fine-tuned with real and mixed data. We also report joint object localization and viewpoint estimation results (AAVP Chapter 6 measure). As in Chapter 6, for viewpoint prediction we rely on a regressor trained on convnet features fine-tuned for detection.

We observe that the texture renderings improve performance on all models (e.g. VGG16 58.8 to 61.9 mAP). Combining the three models further improves detection performance achieving state-of-the-art viewpoint estimation. Adding size specific VGG16 models (like in subsection 10.6.1) further pushes the results, improving up to 5 mAP on small/medium sized objects. Adding bounding box regression, our final combination achieves 67.2 mAP, the best reported result on Pascal3D+.

10.9 CONCLUSION

In this chapter, we presented new results regarding the performance and potential of the R-CNN architecture. Although higher overall performance can be reached with deeper convnets (VGG16), the considered state-of-the-art networks have similar weaknesses; they underperform for truncated, occluded and small objects (§10.6). Additional data does not solve these weak points, hinting that structural changes are needed. Despite common belief, our results suggest these models are not invariant to various appearance factors. Increased training data, however, does improve overall performance, even when using synthetic image renderings (§10.7).

In future work, we would like to extend the CAD model set in order to cover more categories. Understanding which architectural changes will be most effective to handle truncation, occlusion, or small objects remains an open question.

CONCLUSIONS 11

Contents

Understanding visual scenes as a whole, at human quality level or even better, has been a long standing goal of computer vision research (Marr and Nishihara, 1978; Lowe, 1987; LeCun *et al.*, 2015). Due to the sheer complexity of this problem, it has been decomposed into well defined sub-tasks, like object recognition, detection, tracking and segmentation. As objects play a central role in the visual world, robust object representations have been considered a most prominent unit in all visual tasks. Driven by the promise of bridging the gap between standard object class representations on the one hand, and high-level visual understanding tasks on the other hand, richer object representations (Aubry *et al.*, 2014; Deng *et al.*, 2013; Xiang *et al.*, 2014b; Geiger *et al.*, 2014; Carreira and Sminchisescu, 2012) have been coming into focus. Building more expressive object representations is challenging due to three major difficulties. First of all, these richer representations have to deliver more detailed object hypotheses, reliably describing 3D object properties (Xiang and Savarese, 2012; Fidler *et al.*, 2012) like 3D shape, 3D viewpoint, 3D position, metric size and 3D parts, or reason about object context (Tang *et al.*, 2012) or its interaction with other objects (Yang *et al.*, 2012). Second, these representations need to retain high quality levels (Xiang *et al.*, 2015b) in addition to the efficient matching to image evidence (Girshick, 2015). Richer object representations have to be able to deliver excellent quality in challenging realistic scenarios, where the number of object categories is high and object appearance varies drastically due to variation in illumination, viewpoints, context, shapes, etc. Third, the tedious annotation process results in scarce amounts of additional labels (3D shapes, viewpoints, fine-grained categories), posing serious training challenges. Therefore, learning richer representations has to resort to combining different data modalities (e.g. CAD models and real images) or to reusing and sharing information across the data (e.g. via knowledge transfer). In this thesis, we have investigated

these challenges in different contexts.

First, motivated by the three-dimensional nature of objects, we have explored learning 3D object representations. In the context of deformable parts models, we have designed a palette of multi-view and 3D DPMs, representing either discrete or continuous object viewpoints and modeling parts in 3D. These methods, by learning robust object appearance from real images, and descriptive object geometry from CAD data, have demonstrated excellent detection quality, both in terms of object localization and viewpoint estimation on several challenging detection benchmarks.

Second, inspired by the goal of detailed 3D shape representations, we have investigated 3D object class detection in challenging real world scenarios. By aligning CAD models to objects in images, the resulting method can estimate the 3D shape, viewpoint and position of objects. Relying on convnets for 2D localization, viewpoint and keypoints detection, this 3D object detection method provides state-of-the-art 3D object detection performance in realistic (Pascal3D+) scenarios.

Third, we have demonstrated that fine-grained representations can be beneficial for higher-level tasks. In the context of 3D scene understanding, we have shown that metric information, obtained by providing fine-grained categorization of detected objects can in turn be used for tighter and more accurate 3D localization of objects. In the context of general object class detection, we have shown that fine-grained and multi-view representations together result in improved overall detection performance. At the same time, due to the sparse viewpoint data for fine-grained categories, we empirically verified that knowledge transfer techniques can be successfully used to learn dense multi-view models from sparse viewpoint data.

Fourth, driven by the fact that occlusions are not random, we have designed occlusion-aware object representations. These representations capture characteristic object occlusion patterns and have been shown to deliver excellent detection performance even when highly occluded objects are encountered. The occlusion-aware detector won the object detection benchmark in the RMRC 2013 challenge.

And fifth, in addition to building richer representations, we have investigated how convnets cope with appearance variation due to factors like viewpoint, shape, size and partial objects. We first focused on understanding what have state-of-the-art architectures learned in the context of object class detection. After realizing the common weaknesses of these architectures, in a second step we focused on what can they learn, when we increase and vary the training data distribution.

In summary, this thesis has achieved encouraging results towards richer object representations with respect to the different contexts mentioned previously, namely, multi-view and 3D object representations, fine-grained and occlusion representations and finally towards understanding convnets. In parallel to the detailed and more expressive hypotheses, the richer representations presented in this thesis have achieved outstanding detection performance often improving over the previous state-of-the-art results or being on par with them. Therefore we consider the presented work to be a valuable contribution to the field, bridging the gap between machine and human vision and accelerating the pace towards understanding visual scenes in their entirety. To that end, in the coming section we detail and further discuss

the contributions of this work towards richer object representations. Given the large scope of the topic, a single thesis can not cover and explore all relevant research directions, therefore in the last part of this chapter we provide guidelines towards potential future research directions, first w.r.t. to the individual fields but then also going beyond what has been presented.

11.1 DISCUSSION OF CONTRIBUTIONS

In this thesis, we explored richer object representations in the context of object class detection, in several different directions. The resulting contributions are in object class detection in general, as well as 3D object and fine-grained representations. In the following we summarize them all.

11.1.1 Contributions to 3D object representations

In terms of 3D object representations, we have made several notable contributions. In Chapter 3, we have introduced 3D object geometry to the DPM (Felzenszwalb *et al.*, 2010) in three consecutive steps. First of all, the DPM-VOC+VP (Chapter 3) model has been jointly trained to optimize for object localization and viewpoint estimation in a structured output learning framework, resulting in remarkable viewpoint estimation performance, outperforming standard multi-view object detectors by a significant margin on the 3D object classes dataset. Second, we have introduced CAD data in the model learning as a proxy of 3D object geometry. And third, using the CAD models, we have parameterized object parts in 3D space, allowing each viewpoint specific component to use the subset of parts which are actually visible in the component. This effectively enables the model to establish part correspondences across multiple views of the same object. In fact, we have quantified that the DPM-3D-Constraints model can reliably estimate part correspondences in ultra-wide baseline matching experiment, outperforming the work of Zia *et al.* (2013a) on this task by a large margin.

In Chapter 4 we go one step further and extend the DPM-3D-Constraints model with 3D part parameterization. While DPM-3D-Constraints represents part displacement terms in each viewpoint component independently, in $3D^2PM$ the part displacements are represented in 3D leading to a much more compact model representation. In addition, the $3D^2PM$ model introduces a continuous viewpoint model, representing all possible object viewpoints. This has been achieved by learning a viewpoint basis and expressing each viewpoint as a linear combination of the basis. The continuous viewpoint representation has allowed the model to establish angular accurate viewpoint estimates, leading to state-of-the-art viewpoint estimation performance on the EPFL multi-view car dataset (Ozuysal *et al.*, 2009). At the same time, the full 3D part parameterization has resulted in even more accurate part correspondences than DPM-3D-Constraints, leading to state-of-the-art ultra-wide baseline matching performance.

In Chapter 5 we have continued the investigation of multi-view and 3D DPMs. First of all, we have introduced an unifying view on DPMs as star-shaped conditional random fields. Second, we have demonstrated again the outstanding joint object localization and viewpoint estimation performance of our multi-view object representations, which have achieved state-of-the-art viewpoint estimation performance on Pascal3D+ (Xiang *et al.*, 2014a), and outperformed the DPM on KITTI (Geiger *et al.*, 2012) by large margin. In addition, we have further confirmed the excellent performance of our 3D DPMs, now on several datasets and many more object categories. Most notably, the $3D^2PM$ has outperformed the DPM (Felzenszwalb *et al.*, 2010) in terms of object localization on the challenging KITTI dataset. Thus while being significantly better than any previous 3D object representation in terms of localization and viewpoint estimation, we also demonstrated that $3D^2PM$ has competitive performance to state-of-the-art object class detection methods.

Finally, in Chapter 6 we have demonstrated a 3D object class detection method that works in challenging real world scenarios. The proposed method is a pipeline of carefully designed stages, aligning CAD models to objects in images, resulting in a very rich set of output hypotheses: 3D shape, viewpoint and position. At the core of the 3D object detection method are convnets, which we used in different stages of the pipeline for 2D localization, continuous viewpoint estimation and keypoint localization. The 3D representation in this case consists of a collection of CAD models of the object class of interest. The presented method has shown excellent object localization and viewpoint estimation performance, reaching state-of-the-art results on the Pascal3D+ benchmark. Using segmentation as a proxy task, we have shown decent 3D CAD model alignment quality. The resulting segmentation has been shown to be comparable to native and renowned segmentation methods.

11.1.2 Contributions to object class detection in general

As far as object class detection is concerned, we have presented an occlusion-aware object class representation. We have contributed an occlusion pattern mining method capable of isolating characteristic patterns of object-object occlusions (Chapter 9). Focusing on a driving scenario, the occlusion pattern mining method discovered parked cars as the predominant occlusion pattern. In a subsequent step, we build three different part-based representations, with varying degree of sophistication, capturing both the appearance of the occludee and of the occluder. By explicitly representing the occluder, the presented model has shown remarkable detection performance on the objects with 20% - 80% occlusion on the *Car* and *Pedestrian* categories of the KITTI object detection benchmark (Geiger *et al.*, 2012). At the same time, combining the occlusion-aware object detector with detectors for fully visible objects resulted in state-of-the-art performance on the KITTI detection benchmark at the time, winning the detection benchmark of the Reconstruction Meets Recognition Challenge (Urtasun *et al.*, 2013).

In Chapter 10 we focus on understanding convnet representations from the perspective of object class detection. Starting from the usual assumption that bigger

models and more data always help, we use the three state-of-the-art architectures: AlexNet, GoogleNet and VGG16 to first answer the question "what have convnets learned?". To that end, we realize that indeed bigger models do lead to better performance, however the three architectures share the same weaknesses and are not invariant to many appearance factors. Namely, the state-of-the-art convnets are really good in detecting common, high-to-medium resolution objects, while completely failing when it comes to detecting outliers, low-resolution and partially visible objects. Most notably, increasing the amount of low-resolution and outlier training data does not address these weaknesses, suggesting that architectural changes are needed and/or different learning techniques to handle outliers and low-resolution data. In a next step, we explored "what could convnets learn?" in the light of introducing additional training data sampled from the common cases, concluding that additional training data from the common cases does help the overall performance even when using unrealistic data renderings. As expected, we illustrate that the more realistic the generated data is, the higher the gains in performance. At last, by combining the best practices learned in this work, namely, combining features from the different architectures and having size specific models, we achieve state-of-the-art results on Pascal3D+, outperforming the previous state-of-the-art by 12%.

While object class detection and in particular DPM (Felzenszwalb *et al.*, 2010) and R-CNN (Girshick *et al.*, 2014) have been phrased as one-vs-all class specific learning problems, in chapters 3, 5 we explore model learning in a structured output learning framework, by explicitly encoding the Pascal intersection-over-union localization criterion. While similar in spirit to the one presented by Blaschko and Lampert (2008), we demonstrate that the structured output formulation optimizing for object localization outperforms the standard DPM learning on Pascal VOC 2007 (Everingham *et al.*, 2010), 3D object classes (Savarese and Fei-Fei, 2007) and KITTI (Geiger *et al.*, 2012).

11.1.3 Contributions to fine-grained representations

As far as fine-grained representations are concerned, in Chapter 7, we have explored fine-grained information in context of 3D scene understanding. Namely, we have developed two part-based fine-grained categorization methods, one trained in one-vs-all fashion on subordinate categories, and the other jointly trained in the cross-product of viewpoints and fine-grained categories. The jointly trained representation introduced in this work, has achieved state-of-the-art fine-grained categorization and viewpoint estimation performance on a fine-grained dataset of cars. After confirming that part appearance and geometry encode fine-grained affiliations, in a next step we have demonstrated how the fine-grained labels can be leveraged for tighter and more accurate localization of objects in 3D space. More specifically, we have shown that the detailed metric information results in higher precision when estimating the object distance from camera.

On the other hand, in Chapter 8 we explored fine-grained representations in the context of object class detection. We have demonstrated that multi-view and

fine-grained representations can be further leveraged to improve general object class detection. Challenged with the sparse fine-grained data across viewpoints, we have developed, to the best of our knowledge, the first knowledge transfer technique for multi-view and fine-grained representations. We have proposed two techniques towards learning feature-level prior distributions over permissible multi-view representations, based on sparse and dense correlation structures between cells. With an extensive evaluation, we have demonstrated that multi-view models can be successfully learned from only few viewpoints. We have confirmed the improved performance of the fine-grained and multi-view representations in terms of simultaneous object localization and viewpoint estimation on realistic street scenes dataset (KITTI tracking, Geiger *et al.* (2012)).

11.2 FUTURE PERSPECTIVES

In this section, we first discuss limitations of the presented work and then continue with the potential directions of future improvements. We start with the individual topics explored in this thesis and finish with broader long-term objectives.

11.2.1 3D object representations

Compact and scalable continuous multi-view representations Representing and estimating object viewpoint has played an important role in this thesis. We have presented a large variety of discrete and continuous viewpoint representations, and in our work on multi-view and 3D DPMs we have demonstrated excellent viewpoint performance. While coarse viewpoint estimation has seen tremendous success, fine-grained continuous viewpoint prediction on the other hand, still remains an open question. In addition, estimating viewpoints of partial objects, disambiguating opposite views are difficult problems that have not been addressed in the literature. Given the potential of convnets to solve large palette of problems, we believe that convnets are the way to address these issues. As the discrete treatment of viewpoints does not scale towards finer viewpoint granularity, compact continuous representations, sharing parameters across viewpoints constitute a valid research direction. An additional challenge that needs to be addressed is the lack of large scale and realistic benchmarks with angular accurate labels. Pascal3D+ and KITTI are the largest benchmarks, however, the data and category volume in those benchmarks is orders of magnitude smaller than ImageNet. Therefore, research in this area would benefit from larger benchmarks providing angular accurate viewpoint annotations in realistic scenarios.

Local 3D part mixtures The power of the 3D DPM model comes from its compact representation of parts in 3D, relying on well aligned and fully visible objects to train from. Intuitively, improving the model in general means learning realistic and stronger part representations. As strong, local mixture of parts (Yang *et al.*,

2012; Pishchulin *et al.*, 2013b) have shown tremendous potential in human pose estimation, the same can be translated to part-based 3D object representation. Introducing local part mixtures, representing different appearance modes and characteristic part occlusions would loosen the requirement of visible and well aligned training data.

Joint learning from 3D CAD models and realistic images The multi-view and 3D DPMs presented in this thesis leverage CAD models to learn about object geometry and realistic images to learn realistic appearance representations. At the same time, it is apparent that the CAD data contains useful information that can be used to learn representative object appearance as well. Therefore, it remains an open question how to combine CAD models with real world imagery in order to obtain realistic appearance representations. Since the two data domains are different, domain adaptation techniques are strong candidates for future exploration.

3D DPM and convnets As convnets are state-of-the-art object representations, it seems natural to use them for 3D part-based representations. As Girshick *et al.* (2015) have shown that DPMs can be seen as convnets and Tompson *et al.* (2014) have shown that the parameters of a human pose estimation model, both unaries and pairwise can jointly be learned in a convnet architecture, we believe that 3D DPMs can also be linked and implemented with convnets. Combining the representative power of convnets with the compact nature of 3D DPMs would result in a natural and compact convnet-inspired object representation.

Detailed object representations for 3D alignment The 2D-3D lifting in Chapter 6 is driven by a small set of object keypoints which are annotated in the training set. As this set is coarse, we believe that for finer and more accurate shape predictions a larger set of keypoints is required. As the keypoint annotation process is tedious and troublesome, we believe that the set of keypoints can be increased in an unsupervised way. A larger set of keypoints can be achieved either by unsupervised part discovery on the CAD model surfaces, e.g. by segmenting the meshes and establishing correspondences (Shalom *et al.*, 2008), or by searching for latent discriminative parts. This richer shape representation for 3D lifting would lead to more accurate alignment and thus shape estimation.

Joint 3D alignment and segmentation While increasing the set of keypoints seems intuitive and straightforward, alternatively one could explore appearance based cues to improve 3D alignment. Having a detailed and accurate outline of the 2D shape of the object in the image is highly informative for the 3D lifting. Therefore, we believe joint 3D alignment and object segmentation is an interesting direction to explore in a future work. Each task would help the other, for example by serving as a regularizer in the inference process.

Feed-forward 3D object detection with backwards refinement The 3D object detection method in Chapter 6 is a feed-forward pipeline, with each consecutive stage blindly trusting the preceding stage. If one of the stages goes wrong (e.g. viewpoint prediction), there is no space left for the succeeding stages to recover from that mistake. Therefore, we believe that a natural extension of the 3D object detection pipeline would have to include a feedback loop, iteratively refining the predictions of the previous stages. Towards that direction, a natural model to consider would be recurrent neural nets and long-short term memory (Hochreiter and Schmidhuber, 1997), in particular due to its ability to preserve the long-term relationships in the data.

Joint object localization, camera estimation and shape prediction Finally, the pipeline presented in Chapter 6 could be fully replaced by a joint representation, simultaneously predicting object location, camera parameters and 3D shape. Treating several tasks jointly has proven beneficial in the context of convnets (Girshick, 2015) with the individual tasks benefiting from the joint treatment. Following this line of work, we expect that joint training of convnets for the 3D detection related tasks would result in major improvements in the 3D object detection performance.

11.2.2 General object class detection

Multi-object detection in cluttered scenarios Robust detection of objects in clutter (crowds, parking lots for bicycles and cars) represents one of the major challenges in computer vision. In Chapter 9 we have concentrated on representing at most 2 objects in occlusion interaction, which poses challenges for the detection post-processing step, the non-maxima suppression, especially in cluttered scenarios. Inspired by recent works leveraging contextual object relations for image retrieval (Johnson *et al.*, 2015), object detection (Alexe *et al.*, 2012), people detection (Yang *et al.*, 2012) and cell detection (Arteta *et al.*, 2013), as a future work, we believe that jointly reasoning about multiple objects in cluttered scenarios is a valuable direction to pursue. The main challenges towards this direction would be to have an adaptive and compact model, representing not just the individual objects, but also their mutual interactions, jointly counting, localizing objects and reasoning about their 3D and occlusion properties.

Detailed occlusion representations The most successful occlusion-aware model in Chapter 9 is a coarse object-level representation, capturing the visible portion of the occluded object and the occluding portion of the occluder. Although this strategy has resulted in excellent detection performance, a more detailed occlusion representation, capable of finer occlusion reasoning, e.g. to delineate the occlusion boundaries between objects, would be worthwhile exploring. Following recent advances on occlusions, representing truncations in addition to occlusions (Xiang *et al.*, 2015b), modeling human part presence in images (Desai and Ramanan, 2012), detailed occlusion representations constitute a promising

directions for future research. One possible direction would be to introduce localized mixtures of parts (Yang *et al.*, 2012), representing parts in their characteristic visible poses, as well as characteristic part-level occlusions. Using 3D annotated datasets like KITTI or Pascal3D+ such supervision can be extracted directly from the 3D labels. Alternatively, an interesting direction to explore would be to jointly reason about segmentation, occlusion, occlusion boundaries, similar in spirit to the joint localization and object segmentation work by Fidler *et al.* (2013).

Understanding convnet behavior in highly controlled scenarios In Chapter 10 we have analyzed convnet behavior w.r.t. to various appearance factors, assuming object appearance factorizes across these factors. However, obviously in real world datasets like Pascal3D+ the appearance factors are mutually intertwined and heavily correlated. Therefore, cleaner analysis of convnet behavior in a highly controlled setup could alleviate the drawbacks of real world benchmarks. To that end, since synthetic data renderings have been exploited throughout this thesis, we consider generated synthetic data (Aubry and Russell, 2015; Peng *et al.*, 2014) to constitute a valuable asset for this task. By training and evaluating current convnet architectures on generated imagery, we would hope to reach deeper understanding of the actual limits and model capacity of convnets. In particular, we could precisely point out at which object sizes convnets brake, up to what level of occlusion can they actually work, etc. In addition, synthetically generated data allows to understand the impact of appearance factors which are hard to isolate in real world imagery like shapes, shading and light sources.

Comparing different architectures and convnet layers As pointed out in Chapter 10, bigger models do lead to better object detection in general. The same holds for the performance analysis across appearance factors. On the other hand, keeping object detection performance aside, it remains an open question, how different architectures handle different object properties, such as viewpoints and shapes. In particular, as VGG is significantly better than AlexNet on object detection and recognition, and given the intuition that better detection models tend to be invariant e.g. to viewpoint, would that mean that VGG is worse than AlexNet on e.g. viewpoint prediction. In addition, comparing and understanding the behavior of individual layers in terms of the various appearance factors, across architectures, would constitute a valuable asset in understanding and comparing how different architectures represent and handle visual information.

Handling outliers and low resolution objects with convnets Improving convnets performance on small and/or partially occluded objects remains an open research question. As pointed out in Chapter 10, for the current convnet architectures introducing more training data for these specific cases does not really solve this issue. Therefore we strongly believe that following three directions

are promising. First, convnet architectures specialized for small resolution data, possibly with specialized pooling layers and convolutional filters for small objects, are worth exploring. Second, convnet learning techniques specialized to low-resolution and outlier data, potentially regularizing the learning of outlier data, seem like a very sensible research direction. And third, combining the previous two directions has even higher potential to alleviate these issues.

Exploring data synthesis for object detection A straightforward way to improve convnets is to provide additional training data. While labeling data is a tedious task (especially when it comes to 3D annotations) alternatively, generating data by rendering CAD models could be explored. Although this is straightforward way of thinking, there are many questions on the rendering side that have to be answered. How should the data be generated, how far should it deviate from the real data statistics, what is the minimum degree of realism one should obtain to have satisfactory performance, are relevant questions. Furthermore, the properties of the rendering pipeline itself, in terms of shading, material, texture, illumination and background representation and their impact on learning high quality models is the key towards addressing this problem. Therefore, we believe that a strong analysis on data synthesis techniques from the perspective of convnets is necessary in order to leverage synthetically generated data and further boost the field.

Domain adaptation and convnets Generating additional training data typically results in statistical deviations from real data. Hence the domain shift, which can be handled with domain adaptation techniques. Due to the increased interest in generating training data, and the evident domain gap, we consider domain adaptation to be one of the key components to scaling up and boosting convnets for detection and other vision tasks.

Structured output learning of convnets Convnet learning is typically phrased as a classification problem, relying on a softmax loss. As convnets have been trained to abstract away from spatial details (Chen *et al.*, 2015), it becomes even more important to explicitly train them for the task at hand, e.g. object localization. To that end, we are strongly assured that explicitly addressing object localization during convnet learning, similar in spirit to the structured output learning method of (Blaschko and Lampert, 2008), is a very promising research direction.

11.2.3 Fine-grained recognition

Scalable fine-grained representations The jointly trained multi-view and fine-grained representation in Chapter 7 becomes prohibitive, both in terms of training and test complexity, due to the large model size. Therefore, the model can hardly be employed at scale. To that end, alternative scalable fine-grained representations should be explored. As fine-grained categories are hierarchical in nature,

forming clusters of similar classes, we believe a hierarchical fine-grained representation (Gao and Koller, 2011), following a pre-defined semantic hierarchy like WordNet, or automatically learned one (Salakhutdinov *et al.*, 2011), sets the path towards scalable fine-grained representations. In particular, the combination of hierarchical fine-grained representations with deep neural networks provides an interesting and promising direction of future research.

Facilitating fine-grained information Our work has leveraged fine-grained representations for 3D scene understanding and multi-view object detection. However, typically fine-grained recognition methods approach this task in isolation, without further facilitating the additional fine-grained information in other vision tasks. As our research and a few other works (Mottaghi *et al.*, 2015; Kar *et al.*, 2015) have shown, fine-grained information is a valuable asset in many applications. To that end, we strongly believe that fine-grained information should be more present and heavily facilitated in many vision tasks like object tracking, segmentation, alignment. The added level of detail allows for highly interpretable and specialized models, which we believe is a promising direction towards increasing model precision.

Exploring different sources for multi-view knowledge transfer The multi-view priors in Chapter 8 are learned directly from category level visual data, relying on the viewpoint information provided in the dataset. Obviously this works for very related object categories like vehicles, however going towards higher levels in the object hierarchy requires more sophisticated category alignments. Therefore, we consider other sources of category relatedness as a valuable potential direction for future research on multi-view knowledge transfer. In particular, besides the well established attributes and part-based category embeddings for knowledge transfer (Rohrbach *et al.*, 2010), we also think that alternative, textual descriptions of geometric object similarity could be exploited. Additionally, 3D geometric representations (CAD data) could be further leveraged to automatically learn about potential correlations across categories, e.g. by building hierarchies of geometrically related sub-structures.

Knowledge transfer with only a few training examples The analysis in Chapter 8 has revealed that knowledge transfer can be successfully applied in real world scenarios when roughly a dozen training examples are available per category, however when only a few examples are provided the knowledge transfer is not as effective. Therefore, knowledge transfer for k-shot object detection scenario is an unsolved and important problem with high potential of strong impact on the computer vision community. Such models would require stronger regularization during learning, and scalable and adaptive means of reusing information across categories.

This is especially appealing in the context of convnet learning. Current deep learning research has left a wide gap in the space of learning high quality representations from a few or no training examples at all. Therefore, from a

scientific point of view, we consider few-shot learning of deep architectures to be a valuable potential direction.

11.3 THE BIGGER PICTURE

While in the previous sections we have discussed future work tightly related to the contributions in the thesis, this section outlines the broader, long-term challenges towards richer object representations and understanding visual scenes as a whole.

3D shape representations at scale Estimating objects 3D shape and pose has been a long standing goal in computer vision. At the same time, current computer vision research strives towards large scale benchmarks like ImageNet. Therefore, building 3D shape representations for large scale benchmarks like ImageNet is an open and relevant problem for future research. Current methods, employing 3D shape representations, are still limited to only a few categories (cars, chairs) and fully visible, high resolution objects. The main challenges remain in the tedious data annotation process, as well as the troublesome matching of 3D models (e.g. CAD data collections) and 2D images.

Therefore, and especially that now tremendous advances in problems like object recognition have been made, we believe that research should focus on richer 3D shape representations which are natural, flexible, scalable and efficient. There are three key aspects towards this goal. First, the shape representation should capture information at different levels of detail, from coarse object-level descriptions, to very specialized fine-grained shape characteristics, e.g. following recent advances on reconstructing chairs from single images (Huang *et al.*, 2015). Second, large and diverse data is crucial component. As noted previously, 3D annotation is tedious, while at the same time, online 3D and 2D repositories with vast amounts of data are available. Therefore, we strongly believe that research should focus on leveraging the two data sources simultaneously, leveraging the complementary appearance and geometric cues in both modalities. Last, efficient and accurate 3D shape estimation of large amount of object categories in the wild is posing serious inference challenges. In that direction, hierarchical representations (Gao and Koller, 2011), combined with branch-and-bound (Sun *et al.*, 2012a,b; Lehmann *et al.*, 2011) inference, are a promising direction for future exploration.

Understanding visual worlds in the wild Visual scene understanding is considered to be the holy grail of computer vision. The key towards understanding scenes are flexible scene representations, capturing the diverse visual information. The visual world contains information on different levels of granularity, from scene-level information like scene geometry to discriminative object-level details like parts. In that sense, current research is limited in different ways. Given the large number of object categories in the visual world, maintaining a flat object representation, with a separate model for each object category,

seems unrealistic in the long term. Instead, hierarchical object representations, sharing and reusing information among related entities. At the same time, this object-level representation has to communicate with the higher, scene level representations, constraining the space of plausible categories and scene types. Second, current state-of-the-art methods in computer vision are feed-forward models, unable to recover from previous mistakes. In addition, current inference techniques typically explore uniformly the input signal (e.g. single image), although the information is typically not uniformly distributed. A scene understanding method should be "smarter" in prediction and be capable of recovering from possible mistakes, given observations in subsequent stages. That requires a representation capable of remembering it's previous decisions in the short and the long term. As the input signals are not uniform, representations and inference should dedicate resources according to the amount of information in different parts of the input signal. Given all these requirements, reinforcement learning, recurrent neural networks combined with attention and memory mechanisms seem an obvious choice for understanding scenes in the wild.

Beyond single images Finally, both scalable 3D representations and 3D scene understanding would benefit from additional evidence going beyond single images. Stereo data, video sequences, RGB-D, geographical information would provide additional cues potentially improving over the single image case by large margin. Reasoning about occlusions, poses and general scene level properties like geometry would be significantly simplified. At the same time, there are many challenges arising with these additional cues, both representational but also computational. One would expect that the different data modalities would be complementary in nature. Multi-model representations should be able to capture this complementarity and avoid representing redundant information, leading to compact representations. Computational efficiency is another important aspect when dealing with multiple modalities. Following the principles from the previous paragraphs, in terms of hierarchical representations, attention mechanisms in combination with the assumed complementarity across modalities seem as a promising direction towards addressing the computational burden.

LIST OF FIGURES

LIST OF TABLES

BIBLIOGRAPHY

P. Agrawal, R. Girshick, and J. Malik (2014). Analyzing the Performance of Multilayer Neural Networks for Object Recognition, in *European Conference on Computer Vision (ECCV) 2014*. Cited on pages 30, 33, 160, and 165.

T. Ahonen, A. Hadid, and M. Pietikainen (2006). Face Description with Local Binary Patterns: Application to Face Recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 26.

B. Alexe, N. Heess, Y. W. Teh, and V. Ferrari (2012). Searching for objects driven by context, in *Advances in Neural Information Processing Systems (NIPS) 2012*. Cited on page 178.

Andrieu, de Freitas, Doucet, and Jordan (2003). An Introduction to MCMC for Machine Learning, *MACHLEARN: Machine Learning*. Cited on page 37.

M. Andriluka, S. Roth, and B. Schiele (2008). People-Tracking-by-Detection and People-Detection-by-Tracking, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2008*. Cited on page 22.

M. Andriluka, S. Roth, and B. Schiele (2009). Pictorial structures revisited: People detection and articulated pose estimation, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009*. Cited on pages 19, 24, 84, and 146.

M. Andriluka, S. Roth, and B. Schiele (2012). Discriminative appearance models for pictorial structures, *International journal of computer vision*. Cited on page 19.

M. Andriluka, L. Sigal, and M. Black (2011). *Benchmark Datasets for Pose Estimation and Tracking*, Springer. Cited on page 24.

M. Arie-Nachimson and R. Basri (2009). Constructing Implicit 3D Shape Models for Pose Estimation, in *International Conference on Computer Vision (ICCV) 2009*. Cited on pages 36, 106, and 132.

C. Arteta, V. S. Lempitsky, J. A. Noble, and A. Zisserman (2013). Learning to Detect Partially Overlapping Instances, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on pages 27 and 178.

L. E. Atlas, T. Homma, and R. J. M. II (1988). An Artificial Neural Network for Spatio-Temporal Bipolar Patterns: Application to Phoneme Classification, in *Advances in Neural Information Processing Systems (NIPS) 1988*. Cited on page 2.

M. Aubry, D. Maturana, A. Efros, B. Russell, and J. Sivic (2014). Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models, in

IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2014. Cited on pages 3, 41, 42, 46, 106, 107, 111, 112, and 171.

M. Aubry and B. C. Russell (2015). Understanding deep features with computer-generated imagery, in *International Conference on Computer Vision (ICCV) 2015*. Cited on pages 31, 33, and 179.

Y. Aytar and A. Zisserman (2011). Tabula rasa: Model transfer for object category detection, in *International Conference on Computer Vision (ICCV) 2011*. Cited on page 132.

H. Azizpour and I. Laptev (2012). Object Detection Using Strongly-Supervised Deformable Part Models, in *European Conference on Computer Vision (ECCV) 2012*. Cited on page 24.

D. H. Ballard (1987). Generalizing the Hough Transform to Detect Arbitrary Shapes, in *RCV 1987*. Cited on page 22.

S. Bao and S. Savarese (2011). Semantic Structure from Motion, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2011*. Cited on pages 3, 52, 66, 82, 124, 126, and 146.

S. Y.-Z. Bao, Y. Xiang, and S. Savarese (2012). Object Co-detection, in *European Conference on Computer Vision (ECCV) 2012*. Cited on pages 35 and 99.

Y. Bao, M. chandraker, Y. Lin, and S. Savarese (2013). Dense Object Reconstruction Using Semantic Priors, in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on page 41.

A. Bar-Hillel and D. Weinshall (2006). Subordinate class recognition using relational object models, in *Advances in Neural Information Processing Systems (NIPS) 2006*. Cited on page 119.

H. B. Barlow (1972). Single units and sensation: A neuron doctrine for perceptual psychology?, *Perception*. Cited on page 9.

J. T. Barron and J. Malik (2015). Shape, Illumination, and Reflectance from Shading, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 41.

J. Behley, V. Steinhage, and A. B. Cremers (2013). Laser-based Segment Classification Using a Mixture of Bag-of-Words, in *International Conference on Intelligent Robots and Systems (IROS) 2013*. Cited on page 157.

S. Belongie, J. Malik, and J. Puzicha (2000). Shape Context: A New Descriptor for Shape Matching and Object Recognition, in *Advances in Neural Information Processing Systems (NIPS) 2000*. Cited on pages 2, 19, 24, and 28.

R. Benenson, M. Mathias, T. Tuytelaars, and L. Van Gool (2013). Seeking the strongest rigid detector, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on page 1.

Y. Bengio and O. Delalleau (2011). On the expressive power of deep architectures, in *Algorithmic Learning Theory (ALT) 2011*. Cited on page 160.

T. L. Berg, A. C. Berg, and J. Shih (2010). Automatic Attribute Discovery and Characterization from Noisy Web Data, in *European Conference on Computer Vision (ECCV) 2010*. Cited on page 132.

M. Blaschko and C. Lampert (2008). Learning to Localize Objects with Structured Output Regression, in *European Conference on Computer Vision (ECCV) 2008*. Cited on pages 54, 55, 70, 88, 175, and 180.

T. W. Bo Li and S.-C. Zhu (2014). Integrating Context and Occlusion for Car Detection by Hierarchical And-Or Model, in *European Conference on Computer Vision (ECCV) 2014*. Cited on page 105.

D. Borrmann, J. Elseberg, K. Lingemann, and A. Nüchter (2011). The 3D Hough Transform for Plane Detection in Point Clouds: A Review and a New Accumulator Design, *3D Research (3DR)*. Cited on page 46.

L. Bourdev and J. Malik (2009). Poselets: Body part detectors trained using 3D human pose annotations, in *International Conference on Computer Vision (ICCV) 2009*. Cited on pages 48 and 147.

S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and S. Belongie (2010). Visual Recognition with Humans in the Loop, in *European Conference on Computer Vision (ECCV) 2010*. Cited on pages 47, 119, and 120.

C. Bregler, A. Hertzmann, and H. Biermann (2000). Recovering Non-Rigid 3D Shape from Image Streams, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2000*. Cited on page 41.

L. Breiman (2001). Random Forests, *Machine Learning (ML)*. Cited on pages 2 and 48.

R. A. Brooks (1981). Symbolic Reasoning Among 3-D Models and 2-D Images, *Artificial Intelligence (AI)*. Cited on pages 2, 6, 8, 21, 51, 66, 82, and 105.

M. C. Burl and P. Perona (1996). Recognition of Planar Object Classes, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 1996*. Cited on page 25.

M. C. Burl, M. Weber, and P. Perona (1998). A Probabilistic Approach to Object Recognition Using Local Photometry and Global Geometry, in *ECCV 1998*. Cited on page 25.

J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu (2012). Semantic Segmentation with Second-Order Pooling, in *European Conference on Computer Vision (ECCV) 2012*. Cited on page 117.

J. Carreira and C. Sminchisescu (2012). CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 45 and 171.

K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman (2014). Return of the Devil in the Details: Delving Deep into Convolutional Nets, in *British Machine Vision Conference (BMVC) 2014*. Cited on pages 160, 161, and 162.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille (2015). Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs, in *ICLR 2015*. Cited on page 180.

X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille (2014). Detect What You Can: Detecting and Representing Objects using Holistic Models and Body Parts, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 24, 26, and 110.

X. Chen and A. Yuille (2014). Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations, in *Advances in Neural Information Processing Systems (NIPS) 2014*. Cited on page 160.

W. Choi, Y. Chao, C. Pantofaru, and S. Savarese (2013a). Understanding Indoor Scenes using 3D Geometric Phrases, in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition 2013*. Cited on page 45.

W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese (2014). Discovering Groups of People in Images, in *European Conference on Computer Vision (ECCV) 2014*. Cited on page 28.

W. Choi, C. Pantofaru, and S. Savarese (2013b). A General Framework for Tracking Multiple People from a Moving Camera, *Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 28.

W. Choi and S. Savarese (2012). A Unified Framework for Multi-Target Tracking and Collective Activity Recognition, in *ECCV 2012*. Cited on page 28.

W. Choi and S. Savarese (2014). Understanding Collective Activities of People from Videos, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. Cited on page 28.

C. B. Choy, M. Stark, S. Corbett-Davies, and S. Savarese (2015). Enriching Object Detection with 2D-3D Registration and Continuous Viewpoint Estimation, in *Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 43.

C. Cortes and V. Vapnik (1995). Support-Vector Networks, *Machine Learning (ML)*. Cited on pages 2 and 28.

N. Dalal and B. Triggs (2005). Histograms of Oriented Gradients for Human Detection, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2005*. Cited on pages 2, 9, 19, 28, 54, 66, 69, 86, 122, 124, 125, 133, 146, and 192.

T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik (2013). Fast, Accurate Detection of 100,000 Object Classes on a Single Machine, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on page 24.

A. P. Dempster, N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society, Series B*. Cited on page 38.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). ImageNet: A Large-Scale Hierarchical Image Database, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009*. Cited on pages 2, 108, 109, and 161.

J. Deng, J. Krause, and L. Fei-Fei (2013). Fine-Grained Crowdsourcing for Fine-Grained Recognition, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on pages 47, 48, and 171.

C. Desai and D. Ramanan (2012). Detecting Actions, Poses, and Objects with Relational Phraselets, in *European Conference on Computer Vision (ECCV) 2012*. Cited on pages 26 and 178.

S. Dickinson, A. Pentland, and A. Rosenfeld (1992). From Volumes to Views: An Approach to 3-D Object Recognition, *Computer Vision, Graphics, and Image Processing: Image Understanding*. Cited on pages 21, 38, and 39.

P. Dollár, R. Appel, S. Belongie, and P. Perona (2014). Fast Feature Pyramids for Object Detection, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 40.

A. Dosovitskiy and T. Brox (2015). Inverting Convolutional Networks with Convolutional Networks, *CoRR*. Cited on pages 31 and 33.

C. Dubout and F. Fleuret (2012). Exact Acceleration of Linear Object Detectors, in *European Conference on Computer Vision (ECCV) 2012*. Cited on page 24.

M. Eichner and V. Ferrari (2010). We Are Family: Joint Pose Estimation of Multiple Persons, in *European Conference on Computer Vision (ECCV) 2010*. Cited on page 27.

A. Ess, B. Leibe, K. Schindler, and L. V. Gool. (2009). Robust Multi-Person Tracking from a Mobile Platform, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 52, 66, 82, and 105.

M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman (2010). The Pascal Visual Object Classes (VOC) Challenge, *International Journal of Computer Vision (IJCV)*. Cited on pages 23, 34, 36, 51, 81, 94, 95, 107, 108, 112, 113, 114, 119, 121, 146, 158, and 175.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman (2007). *The PASCAL VOC 2007 Results*. Cited on pages 2, 14, 53, 58, 59, 61, 62, 72, 75, 78, 95, 161, 185, 186, and 191.

M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool (2006). *The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results*. Cited on page 140.

T. Evgeniou, C. A. Micchelli, and M. Pontil (2005). Learning Multiple Tasks with Kernel Methods, *Journal of Machine Learning Research (JMLR)*. Cited on pages 133 and 134.

R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis (2011). Birdlets: Subordinate Categorization Using Volumetric Primitives and Pose-Normalized Appearance, in *International Conference on Computer Vision (ICCV) 2011*. Cited on pages 48 and 119.

L. Fei-Fei, R. Fergus, and P. Perona (2006). One-Shot Learning of Object Categories, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 124 and 132.

P. F. Felzenszwalb, R. B. Girshick, and D. McAllester (2009). *Discriminatively Trained Deformable Part Models, Release 4, http://people.cs.uchicago.edu/ pff/latent-release4/*. Cited on pages 59, 85, 95, 124, and 137.

P. F. Felzenszwalb and D. P. Huttenlocher (2005). Pictorial Structures for Object Recognition, *International Journal of Computer Vision (IJCV)*. Cited on pages 5, 10, 24, 84, and 110.

P. F. Felzenszwalb and D. P. Huttenlocher (2012). Distance Transforms of Sampled Functions, *Theory of Computing*. Cited on page 23.

P. F. Felzenszwalb, R. B. rGirshick, D. McAllester, and D. Ramanan (2010). Object Detection with Discriminatively Trained Part Based Models, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 2, 8, 11, 16, 19, 23, 26, 27, 32, 34, 37, 38, 39, 45, 46, 51, 52, 53, 54, 55, 66, 67, 68, 71, 81, 83, 84, 85, 86, 94, 104, 107, 110, 111, 113, 114, 116, 120, 121, 122, 124, 132, 133, 134, 137, 138, 146, 147, 149, 150, 152, 154, 155, 156, 158, 173, 174, 175, 185, 186, and 187.

R. Fergus, P. Perona, and A. Zisserman (2003). Object Class Recognition by Unsupervised Scale-Invariant Learning, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2003*. Cited on pages 25, 51, 66, 81, and 110.

V. Ferrari, T. Tuytelaars, and L. J. V. Gool (2004). Integrating Multiple Model Views for Object Recognition, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2004*. Cited on page 34.

S. Fidler, S. Dickinson, and R. Urtasun (2012). 3D Object Detection and Viewpoint Estimation with a Deformable 3D Cuboid Model, in *Advances in Neural Information Processing Systems (NIPS) 2012*. Cited on pages 38, 39, 45, 46, 106, 132, and 171.

S. Fidler, R. Mottaghi, A. L. Yuille, and R. Urtasun (2013). Bottom-Up Segmentation for Top-Down Detection, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on pages 19 and 179.

P. Fischer, A. Dosovitskiy, and T. Brox (2014). Descriptor matching with convolutional neural networks: a comparison to sift, *arXiv 1405.5769*. Cited on page 160.

P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox (2015). FlowNet: Learning Optical Flow with Convolutional Networks, in *Arxiv 2015*. Cited on page 160.

M. A. Fischler and R. A. Elschlager (1973). The Representation and Matching of Pictorial Structures, *IEEE Trans. Computer*. Cited on pages 21, 24, 25, and 185.

Y. Freund and R. Schapire (1997). A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences*. Cited on pages 2, 24, and 28.

A. Furlan, D. Miller, D. G. Sorrenti, L. Fei-Fei, and S. Savarese (2013). Free your Camera: 3D Indoor Scene Understanding from Arbitrary Camera Motion, in *BMVC 2013*. Cited on page 44.

J. Gall and V. Lempitsky (2009). Class-specific Hough forests for object detection, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009*. Cited on page 22.

T. Gao and D. Koller (2011). Discriminative learning of relaxed hierarchy for large-scale visual recognition, in *International Conference on Computer Vision (ICCV) 2011*. Cited on pages 181 and 182.

T. Gao, B. Packer, and D. Koller (2011). A Segmentation-aware Object Detection Model with Occlusion Handling, in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) 2011*. Cited on page 146.

T. Gao, M. Stark, and D. Koller (2012). What Makes a Good Detector? - Structured Priors for Learning from Few Examples, in *European Conference on Computer Vision (ECCV) 2012*. Cited on pages 132, 134, 136, 137, and 138.

S. K. Gehrig and F. Stein (1999). Cartography and Dead Reckoning Using Stereo Vision for an Autonomous Car., in *ICIP (4) 1999*. Cited on page 3.

A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun (2014). 3D Traffic Scene Understanding from Movable Platforms, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 2, 3, 5, 43, 82, 105, and 171.

A. Geiger, P. Lenz, and R. Urtasun (2012). Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on pages 2, 7, 13, 15, 33, 34, 39, 40, 95, 102, 133, 137, 140, 142, 144, 146, 147, 148, 152, 153, 155, 157, 158, 174, 175, 176, 185, 187, and 192.

A. Geiger, C. Wojek, and R. Urtasun (2011). Joint 3D Estimation of Objects and Scene Layout, in *Advances in Neural Information Processing Systems (NIPS) 2011*. Cited on pages 3, 5, 34, 35, 43, 46, 82, and 88.

A. Ghodrati, M. Pedersoli, and T. Tuytelaars (2014). Is 2D Information Enough For Viewpoint Estimation?, in *British Machine Vision Conference (BMVC) 2014*. Cited on page 35.

R. Girshick (2015). Fast R-CNN, *arXiv 1504.08083*. Cited on pages 161, 169, 171, and 178.

R. Girshick, J. Donahue, T. Darrell, and J. Malik (2014). Rich feature hierarchies for accurate object detection and semantic segmentation, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 1, 2, 6, 7, 8, 14, 19, 28, 43, 48, 81, 107, 108, 114, 160, 161, 164, and 175.

R. Girshick, P. Felzenszwalb, and D. McAllester (2011). Object Detection with Grammar Models, in *Advances in Neural Information Processing Systems (NIPS) 2011*. Cited on pages 19, 24, 27, and 32.

R. Girshick, F. Iandola, T. Darrell, and J. Malik (2015). Deformable Part Models are Convolutional Neural Networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (IEEE Conference on Computer Vision and Pattern Recognition (CVPR)) 2015*. Cited on page 177.

D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich (2011). Viewpoint-aware object detection and pose estimation, in *International Conference on Computer Vision (ICCV) 2011*. Cited on pages 22, 36, 61, 66, 72, 73, 74, 75, 78, 82, 99, 101, 103, 109, 131, 138, and 191.

D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich (2012). Viewpoint-aware object detection and continuous pose estimation, *IVC*. Cited on pages 7 and 22.

I. Goodfellow, Q. Le, A. Saxe, and A. Y. Ng (2009). Measuring Invariances in Deep Networks, in *Advances in Neural Information Processing Systems (NIPS) 2009*. Cited on pages 31, 33, and 165.

K. Green, D. Eggert, L. Stark, and K. Bowyer (1995). Generic Recognition of Articulated Objects through Reasoning about Potential Function, *CVIU*. Cited on pages 66 and 82.

G. Griffin, A. Holub, and P. Perona (2007). Caltech-256 Object Category Dataset, Technical report 7694, California Institute of Technology. Cited on page 124.

C. Gu and X. Ren (2010). Discriminative mixture-of-templates for viewpoint classification, in *ECCV'10 2010*. Cited on pages 5, 34, 35, 46, 60, 106, and 138.

A. Gupta, A. A. Efros, and M. Hebert (2010). Blocks World Revisited: Image Understanding Using Qualitative Geometry and Mechanics, in *European Conference on Computer Vision (ECCV) 2010*. Cited on page 52.

A. Gupta, S. Satkin, A. A. Efros, and M. Hebert (2011). From 3D Scene Geometry to Human Workspace, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2011*. Cited on pages 3 and 52.

S. Gupta, R. Girshick, P. Arbelaez, and J. Malik (2014). Learning Rich Features from RGB-D Images for Object Detection and Segmentation, in *European Conference on Computer Vision (ECCV) 2014*. Cited on pages 6 and 9.

B. Hariharan, J. Malik, and D. Ramanan (2012). Discriminative Decorrelation for Clustering and Classification, in *European Conference on Computer Vision (ECCV) 2012*. Cited on pages 41 and 43.

R. I. Hartley and A. Zisserman (2004). *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2nd edn. Cited on page 63.

V. Hedau, D. Hoiem, and D. Forsyth (2010). Thinking Inside the Box: Using Appearance Models and Context Based on Room Geometry, in *European Conference on Computer Vision (ECCV) 2010*. Cited on pages 39, 44, and 45.

V. Hedau, D. Hoiem, and D. A. Forsyth (2009). Recovering the spatial layout of cluttered rooms, in *International Conference on Computer Vision (ICCV) 2009*. Cited on pages 44 and 45.

M. Hejrati and D. Ramanan (2014). Analysis by Synthesis: 3D Object Recognition by Object Reconstruction, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on page 106.

G. E. Hinton, O. Vinyals, and J. Dean (2015). Distilling the Knowledge in a Neural Network, *CoRR*. Cited on page 31.

M. Hoai and A. Zisserman (2013). Discriminative Sub-categorization, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on pages 132 and 141.

S. Hochreiter and J. Schmidhuber (1997). Long Short Term Memory, *Neural Computation*. Cited on page 178.

D. Hoiem, Y. Chodpathumwan, and Q. Dai (2012). Diagnosing error in object detectors, in *European Conference on Computer Vision (ECCV) 2012*. Cited on pages 26 and 164.

D. Hoiem, A. Efros, and M. Hebert (2008). Putting Objects in Perspective, *International Journal of Computer Vision (IJCV)*. Cited on pages 52, 66, and 82.

D. Hoiem and S. Savarese (2011). *Representations and Techniques for 3D Object Recognition and Scene Interpretation*, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers. Cited on page 2.

J. Hosang, R. Benenson, P. Dollár, and B. Schiele (2015). What makes for effective detection proposals?, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. Cited on page 20.

J. Hosang, R. Benenson, and B. Schiele (2014). How good are detection proposals, really?, in *British Machine Vision Conference (BMVC) 2014*. Cited on pages 20 and 110.

E. Hsiao and M. Hebert (2012). Occlusion Reasoning for Object Detection under Arbitrary Viewpoint, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on page 28.

Q. Huang, H. Wang, and V. Koltun (2015). Single-view reconstruction via joint analysis of image and shape collections, *ACM Transactions on Computer Graphics (TOG)*. Cited on pages 42 and 182.

Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell (2014). Caffe: Convolutional Architecture for Fast Feature Embedding, *arXiv preprint arXiv:1408.5093*. Cited on page 109.

Y. Jiang, J. R. Amend, H. Lipson, and A. Saxena (2012). Learning hardware agnostic grasps for a universal jamming gripper, in *International Conference on Robotics and Automation (ICRA) 2012*. Cited on page 5.

J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei (2015). Image Retrieval Using Scene Graphs, in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 178.

T. Kanade (1980). *A Theory of Origami World*. Cited on page 6.

A. Kar, S. Tulsiani, J. Carreira, and J. Malik (2015). Category-Specific Object Reconstruction from a Single Image, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 8, 41, and 181.

L. Karlinsky, M. Dinerstein, D. Harari, and S. Ullman (2010). The chains model for detecting parts by their context, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2010*. Cited on page 27.

N. Kholgade, T. Simon, A. Efros, and Y. Sheikh (2014). 3D Object Manipulation in a Single Photograph using Stock 3D Models, *ACM Transactions on Computer Graphics (TOG)*. Cited on page 42.

M. Kiefel and P. Gehler (2014). Human Pose Estimation with Fields of Parts, in *European Conference on Computer Vision (ECCV) 2014*. Cited on page 25.

B. Kim, P. Kohli, and S. Savarese (2013). 3D Scene Understanding by Voxel-CRF, in *International Conference on Computer Vision (ICCV) 2013*. Cited on pages 3 and 45.

J. Koenderink and A. van Doorn (1979). The Internal Representation of Solid Shape with Respect to Vision, *Biological Cybernetics*. Cited on page 39.

J. Krause, H. Jin, J. Yang, and L. Fei-Fei (2015). Fine-Grained Recognition without Part Annotations, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 49.

J. Krause, M. Stark, J. Deng, and L. Fei-Fei (2013). 3D Object Representations for Fine-Grained Categorization, in *IEEE Workshop on 3D Representation and Recognition (3dRR) 2013*. Cited on page 48.

A. Krizhevsky, I. Sutskever, and G. E. Hinton (2012). ImageNet Classification with Deep Convolutional Neural Networks, in *Advances in Neural Information Processing Systems (NIPS) 2012*. Cited on pages 1, 2, 6, 7, 9, 13, 19, 28, 29, 108, 160, 161, 162, 164, and 185.

T. Lan, M. Raptis, L. Sigal, and G. Mori (2013). From Subcategories to Visual Composites: A Multi-Level Framework for Object Detection, in *International Conference on Computer Vision (ICCV) 2013*. Cited on pages 132 and 141.

S. Lazebnik, C. Schmid, and J. Ponce (2006). Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2006*. Cited on pages 38, 48, and 121.

Q. V. Le, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng (2012). Building high-level features using large scale unsupervised learning, *International Conference on Machine Learning (ICML)*. Cited on page 160.

L. Leal-Taixe, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savaresee (2014). Learning an Image-based Motion Context for Multiple People Tracking, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on page 28.

Y. LeCun (1988). A Theoretical Framework for Backpropagation, in *Proceedings of the 1988 Connectionist Models Summer School (CSS) 1988*. Cited on page 28.

Y. LeCun, Y. Bengio, and G. Hinton (2015). Deep learning, *Nature*. Cited on pages 28 and 171.

Y. L. LeCun, L. Bottou, Y. Bengio, and P. Haffner (1998). Gradient-Based Learning applied to Document Recognition, *Proceedings of IEEE*. Cited on page 28.

D. C. Lee, M. Hebert, and T. Kanade (2009). Geometric reasoning for single image structure recovery, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009*. Cited on page 45.

A. Lehmann, P. Gehler, and L. Van Gool (2011). Branch&Rank: Non-Linear Object Detection, in *British Machine Vision Conference (BMVC) 2011*. Cited on pages 78 and 182.

B. Leibe, A. Leonardis, and B. Schiele (2004). Combined object categorization and segmentation with an implicit shape model, in *Workshop on statistical learning in computer vision, ECCV 2004*. Cited on pages 22, 34, 51, 81, and 185.

B. Leibe, A. Leonardis, and B. Schiele (2008). Robust Object Detection with Interleaved Categorization and Segmentation, *International Journal of Computer Vision (IJCV)*, vol. 77(1–3), pp. 259–289. Cited on pages 19, 66, and 111.

K. Lenc and A. Vedaldi (2015). Understanding image representations by measuring their equivariance and equivalence, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 31, 33, and 159.

C. Li, D. Parikh, and T. Chen (2012). Automatic Discovery of Groups of Objects for Scene Understanding, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on page 27.

H. Li, Y. Li, and F. Porikli (2014). Robust online visual tracking with an single convolutional neural network, in *Asian Conference on Computer Vision (IJCV) 2014*. Cited on page 160.

L.-J. Li, H. Su, E. P. Xing, and L. Fei-Fei (2010). Object Bank: A High-Level Image Representation for Scene Classification and Semantic Feature Sparsification, in *Advances in Neural Information Processing Systems (NIPS) 2010*. Cited on pages 120 and 121.

J. Liebelt and C. Schmid (2010). Multi-View Object Class Detection with a 3D Geometric Model, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2010*. Cited on pages 8, 37, 38, 60, 61, 62, 66, 72, 73, 82, 99, 106, 112, 131, 132, and 138.

J. Liebelt, C. Schmid, and K. Schertler (2008). Viewpoint-independent object class detection using 3D Feature Maps, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2008*. Cited on page 106.

J. J. Lim, A. Khosla, and A. Torralba (2014). FPM: Fine pose Parts-based Model with 3D CAD models, in *European Conference on Computer Vision (ECCV) 2014*. Cited on pages 106, 107, and 112.

J. J. Lim, H. Pirsiavash, and A. Torralba (2013). Parsing IKEA Objects: Fine Pose Estimation, in *International Conference on Computer Vision (ICCV) 2013*. Cited on pages 106, 107, and 112.

J. J. Lim, R. Salakhutdinov, and A. Torralba (2011). Transfer Learning by Borrowing Examples for Multiclass Object Detection, in *Advances in Neural Information Processing Systems (NIPS) 2011*. Cited on page 132.

D. Lin, S. Fidler, and R. Urtasun (2013). Holistic Scene Understanding for 3D Object Detection with RGBD cameras, in *International Conference on Computer Vision (ICCV) 2013*. Cited on page 45.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick (2014). Microsoft COCO: Common Objects in Context, in *European Conference on Computer Vision (ECCV) 2014*. Cited on pages 2, 10, and 185.

C. Liu, A. Schwing, K. Kundu, R. Urtasun, and S. Fidler (2015). Rent3D: Floor-Plan Priors for Monocular Layout Estimation, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 1.

C. Liu, J. Yuen, and A. Torralba (2011). SIFT Flow: Dense Correspondence across Scenes and Its Applications, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 42.

J. Long, E. Shelhamer, and T. Darrell (2015). Fully Convolutional Networks for Semantic Segmentation, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Cited on page 160.

R. J. Lopez-Sastre, C. Redondo-Cabrera, P. Gil-Jimenez, and S. Maldonado-Bascon (2010). *ICARO: Image Collection of Annotated Real-world Objects*, *http://agamenon.tsc.uah.es/Personales/rlopez/data/icaro*. Cited on pages 5 and 34.

R. J. Lopez-Sastre, T. Tuytelaars, and S. Savarese (2011). Deformable Part Models Revisited: A Performance Evaluation for Object Category Pose Estimation, in *ICCV-WS CORP 2011*. Cited on pages 34, 35, 46, 59, 60, 61, 66, 72, 73, 74, 82, 96, 99, 103, 104, 106, 138, 191, and 192.

D. Lowe (2004). Distinctive image features from scale invariant keypoints, *International Journal of Computer Vision (IJCV)*. Cited on pages 2, 9, and 28.

D. G. Lowe (1987). Three-Dimensional Object Recognition from Single Two-Dimensional Images, *Artificial Intelligence (AI)*. Cited on pages 2, 6, 8, 21, 51, 65, 66, 82, 105, and 171.

A. Mahendran and A. Vedaldi (2015). Understanding Deep Image Representations by Inverting Them, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 30, 31, 33, and 160.

S. Maji, L. Bourdev, and J. Malik (2011). Action Recognition from a Distributed Representation of Pose and Appearance, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2011*. Cited on page 119.

S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi (2013). Fine-Grained Visual Classification of Aircraft, Technical report. Cited on pages 10 and 47.

S. Maji and J. Malik (2009). Object detection using a max-margin Hough transform, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009*. Cited on page 22.

J. Malik (1987). Interpreting line drawings of curved objects, *International Journal of Computer Vision (IJCV)*. Cited on page 6.

D. Marr and H. K. Nishihara (1978). Representation and recognition of the spatial organization of three-dimensional shapes, *Proc. R. Soc. Lond. B (RSLB)*. Cited on pages 2, 5, 20, 51, 65, 66, 82, 105, 171, and 185.

D. Meger, C. Wojek, J. J. Little, and B. Schiele (2011). Explicit Occlusion Reasoning for 3D Object Detection, in *British Machine Vision Conference (BMVC) 2011*. Cited on pages 26 and 32.

M.Hejrati and D.Ramanan (2012). Analyzing 3D Objects in Cluttered Images, in *Advances in Neural Information Processing Systems (NIPS) 2012*. Cited on pages 39, 106, and 132.

R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille (2014). The Role of Context for Object Detection and Semantic Segmentation in the Wild, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 3, 28, 117, 118, and 192.

R. Mottaghi, Y. Xiang, and S. Savarese (2015). A Coarse-to-Fine Model for 3D Pose Estimation and Sub-category Recognition, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 11, 49, 50, and 181.

P. K. Nathan Silberman, Derek Hoiem and R. Fergus (2012). Indoor Segmentation and Support Inference from RGBD Images, in *European Conference on Computer Vision (ECCV) 2012*. Cited on pages 2, 6, 44, 45, and 185.

R. Nevatia and T. O. Binford (1977). Description and Recognition of Curved Objects, *Artificial Intelligence (AI)*. Cited on pages 20 and 21.

A. M. Nguyen, J. Yosinski, and J. Clune (2014). Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images, *CoRR*. Cited on page 30.

M. Nilsback and A. Zisserman (2008). Automated flower classification over a large number of classes, in *ICVGIP 2008*. Cited on pages 47, 119, and 120.

H. Noh, S. Hong, and B. Han (2015). Learning Deconvolution Network for Semantic Segmentation, *CoRR*. Cited on page 1.

P. Ott and M. Everingham (2011). Shared parts for deformable part-based models, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2011*. Cited on page 23.

M. Ozuysal, V. Lepetit, and P. Fua (2009). Pose estimation for category specific multiview object localization, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009*. Cited on pages 34, 37, 39, 73, 74, 95, 103, 106, 107, 132, 173, and 192.

G. Pandey, J. R. McBride, and R. M. Eustice (2011). Ford campus vision and lidar data set, *International Journal of Robotics Research (IJRR)*. Cited on page 126.

M. Pandey and S. Lazebnik (2011). Scene recognition and weakly supervised object localization with deformable part-based models, in *International Conference on Computer Vision (ICCV) 2011*. Cited on page 122.

N. Payet and S. Todorovic (2011). From Contours to 3D Object Detection and Pose Estimation, in *International Conference on Computer Vision (ICCV) 2011*. Cited on pages 35, 61, 66, 73, 82, 99, 131, and 138.

M. Pedersoli and T. Tuytelaars (2014). A Scalable 3D HOG Model for Fast Object Detection and Viewpoint Estimation, in *3D Vision (3DV) 2014*. Cited on page 39.

X. Peng, B. Sun, K. Ali, and K. Saenko (2014). Exploring Invariances in Deep Convolutional Neural Networks Using Synthetic Images, *CoRR*. Cited on pages 31, 33, and 179.

A. P. Pentland (1986). Perceptual Organization and the Representation of Natural Form, *Artificial Intelligence (AI)*. Cited on pages 2, 6, 8, 51, 66, 82, and 105.

B. Pepik, P. Gehler, M. Stark, and B. Schiele (2012a). 3DDPM - 3D Deformable Part Models, in *European Conference on Computer Vision (ECCV) 2012*. Cited on page 16.

B. Pepik, M. Stark, P. Gehler, and B. Schiele (2012b). Teaching 3D Geometry to Deformable Part Models, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on page 16.

B. Pepik, M. Stark, P. Gehler, and B. Schiele (2013). Occlusion Patterns for Object Class Detection, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on page 18.

B. Pepik, M. Stark, P. Gehler, and B. Schiele (2014). Multi-View Priors for Learning Detectors From Sparse Viewpoint Data, in *International Conference on Learning Representations (ICLR) 2014*. Cited on page 17.

B. Pepik, M. Stark, P. Gehler, and B. Schiele (2015a). Multi-view and 3D Deformable Part Models, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 16.

B. Pepik, M. Stark, P. Gehler, and B. Schiele (2015b). What is Holding Back Convnets for Detection?, in *German Conference on Pattern Recognition (GCPR) 2015*. Cited on page 18.

H. Pirsiavash and D. Ramanan (2012). Steerable part models, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on page 24.

L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele (2013a). Poselet Conditioned Pictorial Structures, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on page 24.

L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele (2013b). Strong Appearance and Expressive Spatial Models for Human Pose Estimation, in *IEEE International Conference on Computer Vision (ICCV) 2013*. Cited on page 177.

R. Q. Quiroga, L. R. amd G. Kreiman, C. Koch, and I. Fried (2005). Invariant visual representation by single-neurons in the human brain., *Nature*. Cited on page 9.

N. Razavi, J. Gall, P. Kohli, and L. J. V. Gool (2012). Latent Hough Transform for Object Detection, in *European Conference on Computer Vision (ECCV) 2012*. Cited on page 22.

K. Rematas, T. Ritschel, M. Fritz, and T. Tuytelaars (2014). Image-based Synthesis and Re-Synthesis of Viewpoints Guided by 3D Models, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 42, 43, and 46.

M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele (2010). What Helps Where – And Why? Semantic Relatedness for Knowledge Transfer, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2010*. Cited on pages 132 and 181.

E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, and P. B. Braem (1976). Basic objects in natural categories, *Cognitive Psychology*. Cited on page 119.

C. Rother, V. Kolmogorov, and A. Blake (2004). GrabCut -Interactive Foreground Extraction using Iterated Graph Cuts, *ACM Transactions on Computer Graphics (TOG)*. Cited on page 49.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li (2014). ImageNet Large Scale Visual Recognition Challenge, *CoRR*. Cited on page 10.

R. Salakhutdinov, A. Torralba, and J. B. Tennenbaum (2011). Learning to share visual appearance for multiclass object detection, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2011*. Cited on pages 11, 132, and 181.

M. H. Santosh K. Divvala, Alexei A. Efros (2012). How important are 'Deformable Parts' in the Deformable Parts Model?, in *Parts and Attributes Workshop 2012*. Cited on page 32.

S. Savarese and L. Fei-Fei (2007). 3D generic object categorization, localization and pose estimation., in *International Conference on Computer Vision (ICCV) 2007*. Cited on pages 5, 34, 35, 36, 53, 55, 59, 60, 61, 62, 66, 72, 73, 82, 95, 96, 99, 100, 106, 107, 109, 113, 125, 131, 132, 133, 137, 138, 139, 146, 175, 185, 186, 187, 191, and 192.

F. Schroff, D. Kalenichenko, and J. Philbin (2015). Facenet: A unified embedding for face recognition and clustering, *arXiv*. Cited on page 160.

A. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun (2013). Box In the Box: Joint 3D Layout and Object Reasoning from Single Images, in *International Conference on Computer Vision (ICCV) 2013*. Cited on pages 2, 45, and 46.

E. Seemann, M. Fritz, and B. Schiele (2007). Towards Robust Pedestrian Detection in Crowded Image Sequences, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2007*. Cited on page 22.

P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun (2014). OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks, in *International Conference on Learning Representations (ICLR) 2014*. Cited on pages 19, 28, and 81.

S. Shalom, L. Shapira, A. Shamir, and D. Cohen-Or (2008). Part Analogies in Sets of Objects, in *Eurographics Symposium on 3D Object Retrieval (EUROGRAPHICS) 2008*. Cited on page 177.

J. Shotton, T. Sharp, A. Kipman, A. W. Fitzgibbon, M. Finocchio, A. B. 0001, M. Cook, and R. Moore (2013). Real-time human pose recognition in parts from single depth images, *Commun. ACM*. Cited on page 7.

M. Simon, E. Rodner, and J. Denzler (2014). Part Detector Discovery in Deep Convolutional Neural Networks, *CoRR*. Cited on page 159.

K. Simonyan, A. Vedaldi, and A. Zisserman (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps, *International Conference on Learning Representations (ICLR)*. Cited on pages 29, 33, and 160.

K. Simonyan and A. Zisserman (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition, *International Conference on Learning Representations (ICLR)*. Cited on pages 13, 35, 160, and 164.

H. O. Song, S. Zickler, T. Althoff, R. Girshick, M. Fritz, C. Geyer, P. Felzenszwalb, and T. Darrell (2012). Sparselet Models for Efficient Multiclass Object Detection, in *European Conference on Computer Vision (ECCV) 2012*. Cited on page 24.

J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller (2015). Striving for Simplicity: The All Convolutional Net, *International Conference on Learning Representations (ICLR)*. Cited on page 160.

L. Stark, A. Hoover, D. Goldgof, and K. Bowyer (1993). Function-Based Recognition from Incomplete Knowledge of Shape, in *WQV 1993*. Cited on pages 66 and 82.

M. Stark, M. Goesele, and B. Schiele (2009). A Shape-Based Object Class Model for Knowledge Transfer, in *International Conference on Computer Vision (ICCV) 2009*. Cited on page 132.

M. Stark, M. Goesele, and B. Schiele (2010). Back to the Future: Learning Shape Models from 3D CAD Data, in *British Machine Vision Conference (BMVC) 2010*. Cited on pages 7, 8, 25, 34, 35, 37, 53, 56, 57, 58, 60, 66, 71, 72, 82, 84, 90, 106, 131, 132, and 185.

M. Stark, J. Krause, B. Pepik, D. Meger, J. Little, B. Schiele, and D. Koller (2012). Fine-Grained Categorization for 3D Scene Understanding, in *British Machine Vision Conference (BMVC) 2012*. Cited on pages 10, 17, 47, 48, 94, 132, 141, and 146.

H. Su, M. Sun, L. Fei-Fei, and S. Savarese (2009). Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories., in *International Conference on Computer Vision (ICCV) 2009*. Cited on pages 34, 66, 82, and 131.

M. Sun, S. S. Kumar, G. R. Bradski, and S. Savarese (2013). Object detection, shape recovery, and 3D modelling by depth-encoded hough voting., *Computer Vision and Image Understanding (CVIU)*. Cited on page 36.

M. Sun, B. soo Kim, P. Kohli, and S. Savarese (2014). Relating Things and Stuff via Object Property Interactions, *IEEE Trans. on Pattern Analysys and Machine Intelligence (PAMI)*. Cited on page 28.

M. Sun, M. Telaprolu, H. Lee, and S. Savarese (2012a). Efficient and Exact MAP-MRF Inference using Branch and Bound, in *AISTATS 2012*. Cited on page 182.

M. Sun, M. Telaprolu, H. Lee, and S. Savarese (2012b). An efficient branch-and-bound algorithm for optimal human pose estimation, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on pages 25 and 182.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich (2014a). Going deeper with convolutions, *arXiv*. Cited on pages 13, 160, 161, and 164.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus (2014b). Intriguing properties of neural networks, *CoRR*. Cited on page 30.

S. Tang, B. Andres, M. Andriluka, and B. Schiele (2015). Subgraph Decomposition for Multi-Object Tracking, in *Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 1.

S. Tang, M. Andriluka, A. Milan, K. Schindler, S. Roth, and B. Schiele (2013). Learning People Detectors for Tracking in Crowded Scenes, in *International Conference on Computer Vision (ICCV) 2013*. Cited on page 27.

S. Tang, M. Andriluka, and B. Schiele (2012). Detection and Tracking of Occluded People, in *British Machine Vision Conference (BMVC) 2012*. Cited on pages 27, 32, 33, 147, 150, 154, 155, 156, 157, and 171.

S. Tang, M. Andriluka, and B. Schiele (2014). Detection and Tracking of Occluded People, *International Journal of Computer Vision (IJCV)*. Cited on pages 3 and 27.

A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool (2006). Towards Multi-View Object Class Detection, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2006*. Cited on pages 34, 36, 66, 82, and 106.

E. Tola, V. Lepetit, and P. Fua (2010). DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 38.

J. Tompson, A. Jain, Y. LeCun, and C. Bregler (2014). Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation, *CoRR*. Cited on page 177.

A. Torralba and A. A. Efros (2011). Unbiased look at dataset bias, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2011*. Cited on page 163.

S. Tulsiani and J. Malik (2015). Viewpoints and Keypoints, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 5 and 35.

B. Tversky and K. Hemenway (1984). Objects, parts, and categories, *Journal of Experimental Psychology: General*. Cited on page 47.

J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders (2013). Selective Search for Object Recognition, *International Journal of Computer Vision (IJCV)*. Cited on pages 19, 20, 108, 110, and 161.

R. Urtasun, R. Fergus, D. Hoiem, A. Torralba, A. Geiger, P. Lenz, N. Silberman, J. Xiao, and S. Fidler (2013). *Reconstruction Meets Recognition Challenge (RMRC), http://ttic.uchicago.edu/~rurtasun/rmrc/*. Cited on pages 14, 18, and 174.

A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman (2009). Multiple kernels for object detection, in *International Conference on Computer Vision (ICCV) 2009*. Cited on pages 58, 59, 95, 96, and 191.

A. Vedaldi and A. Zisserman (2009). Structured output regression for detection with partial occulsion, in *Advances in Neural Information Processing Systems (NIPS) 2009*. Cited on pages 26 and 146.

S. Vicente, J. Carreira, L. de Agapito, and J. Batista (2014). Reconstructing PASCAL VOC, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 40 and 41.

M. Villamizar, H. Grabner, J. Andrade-Cetto, A. Sanfeliu, L. V. Gool, and F. Moreno-Noguer (2011). Efficient 3D Object Detection using Multiple Pose-Specific Classifiers, in *British Machine Vision Conference (BMVC) 2011*. Cited on page 131.

C. Wah, S. Branson, P. Perona, and S. Belongie (2011). Multiclass Recognition and Part Localization with Humans in the Loop, in *International Conference on Computer Vision (ICCV) 2011*. Cited on pages 119 and 120.

H. Wang, S. Gould, and D. Koller (2010a). Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding, in *European Conference on Computer Vision (ECCV) 2010*. Cited on pages 44, 45, and 52.

J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong (2010b). Locality-constrained Linear Coding for Image Classification, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2010*. Cited on pages 124, 125, 132, and 192.

S. Wang, S. Fidler, and R. Urtasun (2015). Holistic 3D Scene Understanding from a Single Geo-tagged Image, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 3 and 44.

X. Wang, T. X. Han, and S. Yan (2009). An HOG-LBP human detector with partial occlusion handling, in *International Conference on Computer Vision (ICCV) 2009*. Cited on pages 26, 32, and 146.

X. Wang, M. Yang, S. Zhu, and Y. Lin (2013). Regionlets for Generic Object Detection, in *International Conference on Computer Vision (ICCV) 2013*. Cited on pages 2 and 19.

M. Weber, M. Welling, and P. Perona (2000). Unsupervised Learning of Models for Recognition, in *ECCV 2000*. Cited on page 25.

D. Wei, B. Zhou, A. Torralba, and W. T. Freeman (2015). Understanding Intra-Class Knowledge Inside CNN, *CoRR*. Cited on page 31.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona (2010a). Caltech-UCSD Birds 200, Technical report, California Institute of Technology. Cited on page 10.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona (2010b). Caltech-UCSD Birds 200, Technical report, California Institute of Technology. Cited on pages 47 and 119.

P. Wohlhart, M. Donoser, P. M. Roth, and H. Bischof (2012). Detecting Partially Occluded Objects with an Implicit Shape Model Random Field, in *Proc. Asian Conference on Computer Vision (ACCV) 2012*. Cited on pages 22, 23, and 26.

C. Wojek, S. Roth, K. Schindler, and B. Schiele (2010). Monocular 3D Scene Modeling and Inference: Understanding Multi-Object Traffic Scenes, in *European Conference on Computer Vision (ECCV) 2010*. Cited on pages 3, 43, 46, 52, 66, 82, and 105.

C. Wojek and B. Schiele (2008). A Dynamic Conditional Random Field Model for Joint Labeling of Object and Scene Classes, in *European Conference on Computer Vision (ECCV) 2008*. Cited on page 43.

C. Wojek, S. Walk, S. Roth, and B. Schiele (2011). Monocular 3D Scene Understanding with Explicit Occlusion Reasoning, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2011*. Cited on pages 3, 5, 26, 32, and 146.

C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele (2013). Monocular visual scene understanding: Understanding multi-object traffic scenes, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 3 and 43.

C. Wojek, S. Walk, and B. Schiele (2009). Multi-Cue Onboard Pedestrian Detection, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009*. Cited on page 43.

B. Wu and R. Nevatia (2007). Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet Based Part Detectors, *International Journal of Computer Vision (IJCV)*. Cited on page 1.

T. Wu, B. Li, and S. Zhu (2015a). Learning And-Or Models to Represent Context and Occlusion for Car Detection and Viewpoint Estimation, *CoRR*. Cited on page 3.

T. Wu, B. Li, and S.-C. Zhu (2015b). Learning And-Or Models to Represent Context and Occlusion for Car Detection and Viewpoint Estimation, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 27.

Y. Xiang, A. Alahi, and S. Savarese (2015a). Learning to Track: Online Multi-Object Tracking by Decision Making, in *International Conference on Computer Vision (ICCV) 2015*. Cited on page 28.

Y. Xiang, W. Choi, Y. Lin, and S. Savarese (2015b). Data-Driven 3D Voxel Patterns for Object Category Recognition, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 3, 5, 27, 33, 40, 171, and 178.

Y. Xiang, R. Mottaghi, and S. Savarese (2014a). Beyond PASCAL: A Benchmark for 3D Object Detection in the Wild, in *WACV 2014*. Cited on pages 5, 13, 14, 34, 35, 43, 46, 95, 98, 106, 107, 108, 109, 111, 112, 113, 114, 115, 117, 118, 160, 161, 162, 174, and 187.

Y. Xiang and S. Savarese (2012). Estimating the Aspect Layout of Object Categories, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on pages 39, 99, 100, 103, 106, 131, and 171.

Y. Xiang and S. Savarese (2013). Object Detection by 3D Aspectlets and Occlusion Reasoning, in *IEEE Workshop on 3D Representation and Recognition (3dRR) 2013*. Cited on page 40.

Y. Xiang, C. Song, R. Mottaghi, and S. Savarese (2014b). Monocular Multiview Object Tracking with 3D Aspect Parts, in *European Conference on Computer Vision (ECCV) 2014*. Cited on pages 40, 106, and 171.

J. Xiao, A. Owens, and A. Torralba (2013). SUN3D: A Database of Big Spaces Reconstructed Using SfM and Object Labels, in *International Conference on Computer Vision (ICCV) 2013*. Cited on pages 2, 44, and 185.

S. Xie and Z. Tu (2015). Holistically-Nested Edge Detection, *arXiv 1504.06375*. Cited on page 160.

P. Yan, S. Khan, and M. Shah (2007). 3D Model based Object Class Detection in An Arbitrary View, in *International Conference on Computer Vision (ICCV) 2007*. Cited on pages 66 and 82.

Y. Yang, S. Baker, A. Kannan, and D. Ramanan (2012). Recognizing proxemics in personal photos, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on pages 1, 27, 146, 171, 176, 178, and 179.

Y. Yang and D. Ramanan (2013). Articulated Human Detection with Flexible Mixtures of Parts, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 24, 25, and 115.

B. Yao and L. Fei-Fei (2010). Grouplet: a Structured Image Representation for Recognizing Human and Object Interactions, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2010*. Cited on page 27.

B. Yao, A. Khosla, and L. Fei-Fei (2011). Combining Randomization and Discrimination for Fine-Grained Image Categorization, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2011*. Cited on pages 119 and 120.

J. Yao, S. Fidler, and R. Urtasun (2012). Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on page 3.

J. J. Yebes, L. M. Bergasa, and M. Garcia-Garrido (2015). Visual Object Recognition with 3D-Aware Features in KITTI Urban Scenes, *Sensors*. Cited on page 6.

E. Yoruk and R. Vidal (2013). Efficient Object Localization and Pose Estimation with 3D Wireframe Models, in *IEEE Workshop on 3D Representation and Recognition (3DRR) 2013*. Cited on pages 8, 38, 99, and 106.

J. Yosinski, J. Clune, A. M. Nguyen, T. Fuchs, and H. Lipson (2015). Understanding Neural Networks Through Deep Visualization, *CoRR*. Cited on page 29.

C.-N. J. Yu and T. Joachims (2009). Learning Structural SVMs with Latent Variables, in *International Conference on Machine Learning (ICML) 2009*. Cited on pages 14, 70, and 88.

J. Zbontar and Y. LeCun (2015). Computing the Stereo Matching Cost With a Convolutional Neural Network, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 160.

M. D. Zeiler and R. Fergus (2014). Visualizing and Understanding Convolutional Networks, in *European Conference on Computer Vision (ECCV) 2014*. Cited on pages 29 and 33.

M. D. Zeiler, G. W. Taylor, and R. Fergus (2011). Adaptive deconvolutional networks for mid and high level feature learning, in *International Conference on Computer Vision (ICCV) 2011*. Cited on page 29.

N. Zhang, J. Donahue, R. B. Girshick, and T. Darrell (2014). Part-based R-CNNs for Fine-grained Category Detection, *CoRR*. Cited on pages 48 and 49.

B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba (2015). Object Detectors Emerge in Deep Scene CNNs, *CoRR*. Cited on pages 30 and 159.

X. Zhu and D. Ramanan (2012). Face detection, pose estimation, and landmark localization in the wild, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on page 39.

Y. Zhu, R. Urtasun, R. Salakhutdinov, and S. Fidler (2015). segDeepM: Exploiting Segmentation and Context in Deep Neural Networks for Object Detection, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 28.

M. Z. Zia, M. Stark, B. Schiele, and K. Schindler (2013a). Detailed 3D Representations for Object Recognition and Modeling, *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 3, 6, 8, 37, 44, 65, 82, 106, 107, 112, 173, and 185.

M. Z. Zia, M. Stark, and K. Schindler (2013b). Explicit Occlusion Modeling for 3D Object Class Representations., in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on pages 37, 105, and 132.

M. Z. Zia, M. Stark, and K. Schindler (2014a). *High-Resolution 3D Layout from a Single View*. Cited on page 5.

M. Z. Zia, M. Stark, and K. Schindler (2014b). Towards Scene Understanding with Detailed 3D Object Representations., *International Journal of Computer Vision (IJCV)*. Cited on pages 5, 44, and 47.

M. Z. Zia, M. Stark, and K. Schindler (2015). Towards Scene Understanding with Detailed 3D Object Representations, *International Journal of Computer Vision (IJCV)*. Cited on page 6.

M. Z. Zia, M. Stark, K. Schindler, and B. Schiele (2011). Revisiting 3D Geometric Models for Accurate Object Shape and Pose, in *IEEE Workshop on 3D Representation and Recognition (3DRR) 2011*. Cited on pages 37, 38, 44, 53, 58, 61, 63, 64, 66, 73, 78, 99, 119, 132, 138, and 185.

A. Zweig and D. Weinshall (2007). Exploiting Object Hierarchy: Combining Models from Different Category Levels, in *International Conference on Computer Vision (ICCV) 2007*. Cited on page 132.

CURRICULUM VITAE

Bojan Pepik

Date of birth: 25/12/1984 in Strumica, Macedonia

Citizenship: Macedonian

Education:	01/2011 - today	PhD student in computer science, University of Saarland, Germany; supervised by Prof. Dr. Bernt Schiele.
	04/2009 - 09/2010	Computer science graduate school, University of Saarland, Germany
	09/2003 - 06/2008	Diploma(10 semesters) in electrical engineering, major in computer science, control systems and information technology. Faculty for electrical engineering and information technologies, Skopje, Macedonia.
	06/2003	High school graduation, Strumica, Macedonia.
Experience:	01/2011 - today	PhD student with Prof. Dr. Bernt Schiele, Max Planck Institute for Informatics, Saarbrücken, Germany.
	09/2010 - 12/2010	Internship with Prof. Dr. Bernt Schiele, Max Planck Institute for Informatics, Saarbrücken, Germany.
	06/2008 - 02/2009	Developer at the Digital Image Processing Team (DIPTEAM), Skopje, Macedonia.
	10/2008 - 03/2009	Teaching assistant. Faculty for electrical engineering and information technologies, Skopje, Macedonia.
Invited Talk:	09/2015	Interaction of Automated Vehicles with other Traffic Participants Workshop in conjunction with ITSC 2015.

Academic activities:	Reviewer	IJCV (2015); CVIU (2012, 2013, 2014, 2015); ICCV (2015); WACV (2015); CVPR (2016).

PUBLICATIONS

[1] *Teaching 3D Geometry to Deformable Part Models*
Bojan Pepik, Michael Stark, Peter Gehler and Bernt Schiele.
In 2012 IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**), 2012.

[2] *3D^{2}PM-3D Deformable Part Models*
Bojan Pepik, Peter Gehler, Michael Stark and Bernt Schiele.
In 12th European Conference on Computer Vision (**ECCV**) 2012.

[3] *Fine-Grained Categorization for 3D Scene Understanding*
Michael Stark, Jonathan Krause, Bojan Pepik, David Meger, Jim Little, Bernt Schiele and Daphne Koller.
In British Machine Vision Conference (**BMVC**) 2012.

[4] *Occlusion Patterns for Object Class Detection*
Bojan Pepik, Michael Stark, Peter Gehler and Bernt Schiele.
In 2013 IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**), 2013.

[5] *Multi-View Priors for Learning Detectors From Sparse Viewpoint Data*
Bojan Pepik, Michael Stark, Peter Gehler and Bernt Schiele.
In International Conference on Learning Representations (**ICLR**), 2014.

[6] *Multi-view and 3D Deformable Part Models*
Bojan Pepik, Michael Stark, Peter Gehler and Bernt Schiele.
In IEEE Transactions on Pattern Analysis and Machine Intelligence (**PAMI**), 2015.

[7] *3D Object Class Detection in the Wild*
Bojan Pepik, Michael Stark, Peter Gehler, Tobias Ritschel and Bernt Schiele.
In IEEE Workshop on 3D from Single Image in conjunction with (**CVPR**), 2015.

[8] *What is Holding Back Convnets for Detection?*
Bojan Pepik, Rodrigo Benenson, Tobias Ritschel and Bernt Schiele.
In German Conference on Pattern Recognition (**GCPR**), 2015.