Learning to Segment in Images and Videos with Different Forms of Supervision

A dissertation submitted towards the degree
Doctor of Engineering
(Dr.-Ing.)
of the Faculty of Mathematics and Computer Science
of Saarland University

by
Anna Khoreva, M.Sc.

Saarbrücken
December 2017

Day of Colloquium	20th of December, 2017
Dean of the Faculty	Univ.-Prof. Dr. Frank-Olaf Schreyer Saarland University, Germany

Examination Committee

Chair	Prof. Dr. Matthias Hein
Reviewer, Advisor	Prof. Dr. Bernt Schiele
Reviewer	Prof. Richard Szeliski, Ph.D.
Reviewer	Prof. Dr. Thomas Brox
Academic Assistant	Dr. Gerard Pons-Moll

ABSTRACT

Much progress has been made in image and video segmentation over the last years. To a large extent, the success can be attributed to the strong appearance models completely learned from data, in particular using deep learning methods. However, to perform best these methods require large representative datasets for training with expensive pixel-level annotations, which in case of videos are prohibitive to obtain. Therefore, there is a need to relax this constraint and to consider alternative forms of supervision, which are easier and cheaper to collect. In this thesis, we aim to develop algorithms for learning to segment in images and videos with different levels of supervision.

First, we develop approaches for training convolutional networks with weaker forms of supervision, such as bounding boxes or image labels, for object boundary estimation and semantic/instance labelling tasks. We propose to generate pixel-level approximate groundtruth from these weaker forms of annotations to train a network, which allows to achieve high-quality results comparable to the full supervision quality without any modifications of the network architecture or the training procedure.

Second, we address the problem of the excessive computational and memory costs inherent to solving video segmentation via graphs. We propose approaches to improve the runtime and memory efficiency as well as the output segmentation quality by learning from the available training data the best representation of the graph. In particular, we contribute with learning must-link constraints, the topology and edge weights of the graph as well as enhancing the graph nodes - superpixels - themselves.

Third, we tackle the task of pixel-level object tracking and address the problem of the limited amount of densely annotated video data for training convolutional networks. We introduce an architecture which allows training with static images only and propose an elaborate data synthesis scheme which creates a large number of training examples close to the target domain from the given first frame mask. With the proposed techniques we show that densely annotated consequent video data is not necessary to achieve high-quality temporally coherent video segmentation results.

In summary, this thesis advances the state of the art in weakly supervised image segmentation, graph-based video segmentation and pixel-level object tracking and contributes with the new ways of training convolutional networks with a limited amount of pixel-level annotated training data.

ZUSAMMENFASSUNG

In der Bild- und Video-Segmentierung wurden im Laufe der letzten Jahre große Fortschritte erzielt. Dieser Erfolg beruht weitgehend auf starken Appearance Models, die vollständig aus Daten gelernt werden, insbesondere mit Deep Learning Methoden. Für beste Performanz benötigen diese Methoden jedoch große repräsentative Datensätze für das Training mit teuren Annotationen auf Pixelebene, die bei Videos unerschwinglich sind. Deshalb ist es notwendig, diese Einschränkung zu überwinden und alternative Formen des überwachten Lernens in Erwägung zu ziehen, die einfacher und kostengünstiger zu sammeln sind. In dieser Arbeit wollen wir Algorithmen zur Segmentierung von Bildern und Videos mit verschiedenen Ebenen des überwachten Lernens entwickeln.

Zunächst entwickeln wir Ansätze zum Training eines faltenden Netzwerkes (convolutional network) mit schwächeren Formen des überwachten Lernens, wie z.B. Begrenzungsrahmen oder Bildlabel, für Objektbegrenzungen und Semantik/Instanz-Klassifikationsaufgaben. Wir schlagen vor, aus diesen schwächeren Formen von Annotationen eine annähernde Ground Truth auf Pixelebene zu generieren, um ein Netzwerk zu trainieren, das hochwertige Ergebnisse ermöglicht, die qualitativ mit denen bei voll überwachtem Lernen vergleichbar sind, und dies ohne Änderung der Netzwerkarchitektur oder des Trainingsprozesses.

Zweitens behandeln wir das Problem des beträchtlichen Rechenaufwands und Speicherbedarfs, das der Segmentierung von Videos mittels Graphen eigen ist. Wir schlagen Ansätze vor, um sowohl die Laufzeit und Speichereffizienz als auch die Qualität der Segmentierung zu verbessern, indem aus den verfügbaren Trainingsdaten die beste Darstellung des Graphen gelernt wird. Insbesondere leisten wir einen Beitrag zum Lernen mit must-link Bedingungen, zur Topologie und zu Kantengewichten des Graphen sowie zu verbesserten Superpixeln.

Drittens gehen wir die Aufgabe des Objekt-Tracking auf Pixelebene an und befassen uns mit dem Problem der begrenzten Menge von dicht annotierten Videodaten zum Training eines faltenden Netzwerkes. Wir stellen eine Architektur vor, die das Training nur mit statischen Bildern ermöglicht, und schlagen ein aufwendiges Schema zur Datensynthese vor, das aus der gegebenen ersten Rahmenmaske eine große Anzahl von Trainingsbeispielen ähnlich der Zieldomäne schafft. Mit den vorgeschlagenen Techniken zeigen wir, dass dicht annotierte zusammenhängende Videodaten nicht erforderlich sind, um qualitativ hochwertige zeitlich kohärente Resultate der Segmentierung von Videos zu erhalten.

Zusammenfassend lässt sich sagen, dass diese Arbeit den Stand der Technik in schwach überwachter Segmentierung von Bildern, graphenbasierter Segmentierung von Videos und Objekt-Tracking auf Pixelebene weiter entwickelt, und mit neuen Formen des Trainings faltender Netzwerke bei einer begrenzten Menge von annotierten Trainingsdaten auf Pixelebene einen Beitrag leistet.

CONTENTS

1 INTRODUCTION

The human vision system is remarkable at extracting and analyzing visual information. In a blink of an eye, humans are able to fully analyze a scene and separate all the components present. Computer vision can be seen as a field that aims to give a similar, if not better, capability to a machine or computer to translate the set of pixels into useful information.

In computer vision the human ability of visual grouping has been studied as "segmentation", the partitioning of an image or a video sequence into sets of pixels that correspond to "objects" or parts of objects. The grouping process is usually based on bottom up cues such as similarity of pixels in appearance and motion as well as top down input derived from semantic knowledge of objects. Segmentation plays an important role and emerges in many application areas in computer vision, such as scene understanding, activity recognition, video summarization and indexing, and digital entertainment. Even though segmenting image or video data is one of the most basic and long standing problems, it remains far from being solved effectively.

One of the main difficulties in visual grouping is the problem of high variability. A computer vision system is required to generalize well across objects with large variations in appearance, shape, viewpoint, illumination and occlusion. In addition, visual data can be intrinsically ambiguous and cluttered. When doing image or video segmentation humans use much more knowledge than just relying on color information of pixels and can easily generalize from observing a set of objects to recognizing objects that have never been seen before. Trainable methods have proven to be particularly effective for segmentation as they can model some of this knowledge. However, to perform best they require large-scale and domain-specific databases for training.

Deep learning approaches are among trainable methods that have become the de-facto technique in image and video segmentation. Their recent and incredible performance improvements are enabled by significant increase of available training data and faster computing hardware. Compared to preceding approaches, deep networks require less engineering and can learn most components directly from data with fewer assumptions. However, one of their major downsides is their ever growing hunger for more training data. These models typically require pixel-level annotations during training. Acquiring such data is an expensive and time-consuming process. Therefore there is a need to relax this constraint and to consider alternative forms of supervision which are cheaper and faster to obtain.

Throughout this thesis, we aim to develop algorithms to tackle several image and video segmentation problems with different levels of supervision. In the first part of the thesis we address the problem of using *weak supervision for image segmentation*. Instead of training deep networks with full supervision using expensive pixel-

level object mask annotations we propose to use weak annotations, in the form of bounding boxes (i.e., coarse object locations) or image-level labels (i.e., information about which object classes are present), which are far easier to collect. We consider three closely related tasks: object boundary detection (Chapter 3) - since object boundaries can be used to aid to improve the mask estimation of the object; semantic labelling and instance segmentation (Chapters 4 – 5).

The remaining two parts of the thesis consider a broader domain of video segmentation. In the second part we tackle the problem of *learning video segmentation via graphs*. Given the large amount of pixels/superpixels in a video, graph-based approaches tend to be slow and have a large memory footprint. Consequently, to improve efficiency some approaches consider building a graph only over neighboring frames instead of the whole video volume. However, these methods have difficulties capturing long range relationships and ensuring globally consistent segmentation. In contrast, we propose to improve the runtime complexity and memory efficiency as well as the output quality by learning from the available training data the best representation of the graph. We contribute the integration of the learned must-link constraints (Chapter 6), the strategies to learn the topology and edge weights of the graph (Chapter 7) and to the enhancement of the graph nodes, superpixels, themselves (Chapter 8). The proposed techniques help to improve the scalability of graph-based methods as well as enable improved segmentation of long videos.

The third part of the thesis is focused on *pixel-level object tracking via deep networks*. In the literature this task is frequently referred to as semi-supervised video object segmentation. Given a first frame with the object mask annotation, the goal is to accurately segment the same instance in future video frames. Pixel-level object tracking can be difficult to approach via deep learning methods, since in contrast to images a large body of densely, pixel-wise annotated video data is not available for training. We address this limitation by introducing a network architecture which allows training with static images only (Chapter 9) and by generating synthetic data for training from the given first frame annotation (Chapter 10). With the proposed techniques we show that densely annotated consequent video data is not necessary to achieve high-quality temporally coherent video segmentation results.

The rest of this chapter is organized as follows. First we discuss the main challenges towards solving the aforementioned tasks and our contributions to address them in Section 1.1. Next, we provide an outline of the thesis in Section 1.2.

1.1 CONTRIBUTIONS OF THE THESIS

As discussed above, this thesis focuses on three research directions: *image segmentation with weaker forms of supervision*, *graph-based video segmentation* and *pixel-level object tracking*. Essentially, all three directions are concerned with correctly delineating object masks with different levels of supervision during training and test time. In the following we detail the challenges towards these tasks, as well as the contributions this thesis makes to address them.

1.1.1 Image segmentation with weaker forms of supervision

The first target of the thesis is segmenting images without relying on expensive dense pixel-level annotations for training convolutional neural networks (CNNs, convnets). Specifically we want to address weakly-supervised semantic and instance segmentation problems (Chapters 4 – 5) as well as the related task of object boundary detection (Chapter 3). Even in the fully supervised case, segmenting an object is a difficult task, as apart from recognizing and localizing the object, the goal is also to delineate correctly object boundaries. We start with defining the main challenges for image segmentation via convnets as well as the challenges for training a network with weaker forms of supervision.

1.1.1.1 *Challenges*

Expensive pixel-level annotations. To provide supervision for image segmentation or object boundary detection one need to exhaustively delineate all the objects of interest in the image. Such pixel-level mask annotations are costly and do not scale well, becoming a bottleneck to approach new categories of objects that have not been targeted by the existing image datasets. In order to make the training of new object classes affordable, and/or to increase the capacity of the CNN models, there is a requirement in approaches that can work with weaker forms of supervision.

Reduced spatial resolution. Recently fully convolutional networks (FCNs) have shown remarkable results for image segmentation. However, these approaches have limitation of low-resolution prediction. Multiple stages of spatial pooling and convolution strides significantly reduce the final image prediction, losing the finer image structure. A popular approach is to employ DenseCRF post-processing (Krähenbühl and Koltun, 2011) with tuned parameters to refine the network output. This allows to recover small image details that the network might have missed and to better localize object boundaries. Albeit, it is known that DenseCRF is quite sensitive to its parameters and without tuning can easily worsen results.

Imbalanced datasets. Image datasets are often imbalanced, i.e., most training samples belong to a few majority classes, while the minority classes only contain a scarce amount of instances. Furthermore, objects can occur at different scales making per-pixel class distribution even more skewed. Without handling the data imbalance convolutional models tend to be biased toward the majority classes with poor accuracy for the minority classes. This issue is even more present while training a weakly-supervised model as the system is very susceptible to strong inter-class co-occurrences.

Full extent of the object. In order to keep the annotation cost low, recent work considers to employ image labels as the main form of supervision. Image labels provide a constraint on the desired output: if the label is present at least one pixel

in the image must be assigned to that label, if the label is absent no pixel should have that label. However, the object in the image is rarely a single pixel. Therefore learning the full object extent just from images and their labels is a difficult problem. To enforce larger output region size, some prior knowledge is commonly used, e.g. shape or appearance priors.

Noise in the weakly-supervised data. Box-level annotations are cheaper to get but less precise than pixel-level masks. Thus the problem of supervision with bounding boxes can be seen as an issue of input label noise. Intuitively, a large number of images with box annotations should provide enough information to understand which part of the box belongs to the object and which to background. However, even though the existing semantic image datasets are large-scale they do not cover all possible appearance variations of certain semantic categories. In addition, as mentioned above, these datasets usually have a skewed distribution of classes. Therefore, the network has troubles to de-noise the input during training and thus to output correct predictions at test time. The convnet training might benefit from some de-noising strategy of given box annotations, increasing the ratio of correctly labelled pixels (trading off lower recall for higher precision).

1.1.1.2 *Contributions*

The following summarizes the contributions for *learning to segment images with weaker forms of supervision*.

The first contribution is introducing the task of weakly supervised object-specific boundary detection. To the best of our knowledge there is no previous work attempting to learn object boundaries in a weakly supervised fashion, using just bounding box supervision. In Chapter 3 we show on several boundary estimation benchmarks (BSDS (Arbeláez *et al.*, 2011), Pascal VOC12 (Everingham *et al.*, 2015), MS COCO (Lin *et al.*, 2014), and SBD (Hariharan *et al.*, 2011)) that good results can be obtained without the need of instance-wise object boundary annotations and on some benchmarks performance of the fully supervised state of the art is reached.

The second contribution is an approach for generating approximate pixel-level groundtruth for training the network given bounding boxes. We utilize box annotations to generate an initial guess of the object masks with the help of classic computer vision techniques, elaborately fusing box-driven figure-ground segmentation (Rother *et al.*, 2004), object proposals (Pont-Tuset and Gool, 2015) and/or unsupervised image segmentation (Felzenszwalb and Huttenlocher., 2004). To reduce the noise in the generated training labels we introduce the concept of ignore labels: when in doubt of assigning the label to the pixel we mark it as "ignore" and do not use it during the training. We show that, when properly used, "old-school" computer vision methods are a source of surprisingly effective supervision for convnet training. The proposed approach is effectively employed to address closely akin tasks of object boundary detection in Chapter 3 and semantic and instance segmentation in Chapter 4. We demonstrate that the state-of-the-art quality can be reached when

properly generating training labels from bounding boxes, instead of modifying the convnet training procedure as done in previous work (Dai *et al.*, 2015a; Papandreou *et al.*, 2015). To the best of our knowledge, we are the first to address the weakly supervised instance segmentation task.

The third contribution is exploring recursive training of convnets for weakly supervised semantic labelling, where convnet predictions of the previous training round are used as supervision for the next round. The recursive training is enhanced by de-noising the convnet outputs using information provided from the annotated boxes and object priors. We analyze how to reach good quality results via recursive training and discuss the limitations of the approach (Chapter 4).

The fourth contribution is an effective method for training the semantic segmentation network from image-level annotations. Image labels alone can provide the location of discriminative object regions ("object seeds"), but learning the full object extent is challenging. For feeding the object extent we employ a class-agnostic weakly supervised saliency model as an additional source of information. By combining cues from the seeds and saliency the proposed Guided Segmentation network (Chapter 5) achieves state-of-the-art performance. In addition, we compare the effectiveness of different seed methods for the task and analyze the importance of saliency towards the final quality.

Overall, with weaker forms of supervision we are able to achieve high-quality results for image segmentation, reaching comparable quality to the fully supervised regime.

1.1.2 Video segmentation via graphs

Next we address a broader domain - more specifically, videos. Video segmentation is far less researched compared to image segmentation due to increased computational complexity and the inherent difficulties such as camera-motion, occlusions, non-rigid deformations, changes in illumination and perspective.

Given a video sequence the aim is to generate a hierarchical segmentation, grouping regions in space and time with coherent appearance and motion. This can be seen as a first step to interpret the video content and thus can be employed as the base of higher-level computer vision tasks, such as semantic video segmentation, object tracking, scene understanding, and activity recognition.

Modeling video segmentation as a graph partitioning problem has shown to be effective. The graph is constructed over the whole video sequence, where the nodes represent pixels or superpixels, and the edges encode the spatio-temporal structure. Then usually spectral clustering is employed as a partitioning method. However, a few challenges arise when addressing video segmentation via graphs.

1.1.2.1 *Challenges*

Superpixels as graph nodes. Graph based video segmentation often relies on having high quality superpixels as graph nodes to compute features and affinities.

The motivation to employ superpixels as pre-processing stage is two-fold. First, a desirable reduction of the computational complexity is achieved since the number of graph nodes is lowered by several orders of magnitude. And second, a richer and more powerful per-frame representation can be defined than would not be possible over pixels alone.

Superpixels are the starting point for graph partitioning and directly impact the final segmentation quality. Good superpixels for video segmentation should have a good temporal consistency to make matching across time easier and give a high boundary recall to avoid incorrect merges of pixels in the pre-processing stage. Ideally superpixels should form semantic regions and the number of extracted superpixels per-frame should be as few as possible to accelerate overall computation as well as to reduce the chances of segmentation errors. Thus designing a good superpixel method is key for high-quality video segmentation results.

Graph construction. Constructing the graph is a crucial step for ensuring the performance for graph-based video segmentation methods. However, there have been limited efforts for building effective graphs. The most common method is to form edges between superpixels in a certain spatio-temporal neighbourhood and to estimate the edge weights using the combination of different cues such as color, motion and texture. This approach is clearly suboptimal as it does not explore the available training data to ensure the best topological structure of the graph and affinities for the task at hand.

Efficient partitioning. As mentioned above spectral clustering is a common choice of a partitioning method for graph-based approaches, which has proven to be successful for the task (Fragkiadaki and Shi, 2012; Ochs *et al.*, 2014; Keuper and Brox, 2016; Galasso *et al.*, 2014). Spectral methods convince by their globalization effect and their ability to include long-range connections. Albeit, one of the important limitation of spectral techniques is the excessive computational and memory costs. The limits are particularly clear in the case of high-resolution long video sequences (Galasso *et al.*, 2013), restricting the large-scale applicability of spectral methods. This results in the need of constructing smaller and/or sparser graphs in order to be able to process the video with available computational resources.

Learning from available training data. The other difficulty for video segmentation via spectral clustering is learning from the training data. While often a labeled dataset is available, a systematic learning of the affinities used to construct the graph is arduous. In particular, the optimization of the minimizer which yields the segmentation is out of reach, as solving for the clustering objective (normalized cut) itself is a NP-hard problem and its relaxation is non-convex. Therefore a common practice is to just validate a few model parameters (Galasso *et al.*, 2012; Maire and Yu, 2013). This refrains video segmentation methods employing spectral clustering to profit from the available annotated video data.

1.1.2.2 *Contributions*

In the following we present the contributions of the thesis for the *learning to segment videos via graphs*.

The first contribution is learning and integration of must-link constraints into graph-based video segmentation. Must-link constraints specify that two nodes should be assigned into one cluster. By learning must-link constraints we leverage the available training data in order to avoid undesired solutions in graph partitioning. It also allows reducing the size of the problem, while preserving the original objective for all partitions satisfying the must-links. Experimentally, we show that learned must-link constraints improve the efficiency and, in most cases, performance of the considered graph-based video segmentation methods. See Chapter 6 for details.

The second contribution is a graph construction method for video segmentation. We propose to learn both the edge topology and weights of the graph, leveraging the existing training data for video segmentation. We employ different classifiers for learning the affinities between the graph nodes - superpixels - based on their spatial and temporal distance, and then alter the graph structure by selecting the most confident edges according to the scores of learned classifiers. Learning the graph helps to improve both performance as well as efficiency without changing the graph partitioning method, see Chapter 7.

The third contribution is improved superpixels for graph-based video segmentation. We provide an analysis of the effectiveness of different superpixel methods for video segmentation. Experimentally, we demonstrate that boundary-based superpixels are more suitable for the task, as they are more likely to form semantic regions compared to classical superpixel techniques. As the quality of boundary-based superpixels depends directly on the quality of the initial boundary estimates, we propose to improve boundaries by combining image cues with object-level cues, and merging them with temporal cues. By enhancing boundary estimation in video frames, we improve per-frame superpixels, and thus the final video segmentation quality. For details see Chapter 8.

1.1.3 Pixel-level object tracking via CNNs

Pixel-level object tracking, also referred to as semi-supervised video object segmentation, aims to output the mask of an object throughout a video sequence given its groundtruth segmentation in the first frame. Recently, deep learning based approaches have shown good performance for this task. They often utilize fully convolutional networks (FCNs) designed for image segmentation by processing the video on a frame-by-frame basis. However, when moving from images to video new challenges emerge.

1.1.3.1 *Challenges*

Lack of large-scale annotated video data. Superior performance of CNNs is usually enabled by the availability of large-scale annotated datasets. However, pixel-level object tracking can be difficult to approach via convnets, since labelling videos at the pixel level is a laborious task, and the cost of creating a sufficiently large body of densely, pixel-wise annotated video data for training is usually prohibitive. Several video object segmentation datasets exist: DAVIS_{17} (Pont-Tuset *et al.*, 2017), YouTubeObjects (Jain and Grauman, 2014), FBMS (Ochs *et al.*, 2014), and $\text{SegTrack}_{\text{v2}}$ (Li *et al.*, 2013). However, all of them are relatively small-scale in terms of number of videos (up to 150 videos), particularly in comparison with existing image segmentation datasets ($\sim$300k images in MS-COCO (Lin *et al.*, 2014)). Not all of them are densely annotated (e.g. every 10th frame in YouTubeObjects and every 20th frame in FBMS), making integration of temporal context during training harder. Plus each of them contains different types of object annotations (e.g. moving objects in FBMS, 10 semantic categories in YouTubeObjects without separating instances of the same class, single salient object in $\text{SegTrack}_{\text{v2}}$), making it difficult to combine them to train a network. Thus there is a demand to relax the constraint of relying on consequent video data with pixel-level annotations for training.

Problem of domain shift. It has been shown that given test domain, both for image and video data, that the performance of the segmenter highly depends on the domain it was trained on (Hoffman *et al.*, 2016; Chen *et al.*, 2017b). However, the problem of domain shift is much more severe for videos than for images. Humans might follow different approaches when capturing the scene with a video camera compared to taking a photo. In images the objects tend to be fully in focus, while videos usually have objects coming in and out of the frame, being truncated or fully occluded by distractors. Also the distance at which objects are captured varies much more in videos than in images. Differences in compression schemes, color contrast as well as the proficiency of the videographer can affect the quality of the video. All these factors makes the appearance diversity much higher in video data.

Furthermore, in contrast to static images, video frames are temporally coherent. Frames close in time often contain nearly identical samples of the same objects, whereas in image datasets such repetitions rarely occur. As the space of possible appearance variations is very large, each video dataset, being biased to its own specific setting, can cover it only partially. Therefore, when testing on one video dataset and training on another, performance can be surprisingly poor under domain shifts that appear mild to humans. For good performance one should collect a broad range of videos capturing all possible aspects expected to appear and/or being as close as possible to the test domain.

Encoding temporal context. Compared to static images, videos are a richer source of information. A single video often shows multiple views of an object, its various deformations and articulation states. Plus videos also capture the motion of the objects which enables a better segmentation from the background. However,

modelling temporal information in videos is a very challenging problem, especially when CNN-based supervised learning is used. Videos pose both technical and representational challenges.

From a computational perspective, CNNs require a long training time to effectively optimize the millions of parameters of the model. This difficulty is further compounded when extending the connectivity in temporal dimension as the network must process not just one image but multiple video frames at a time. The computational demands are even worse for pixel-level predictions in videos, as the memory load is dramatically increased.

Besides, it is also unclear how to extend successfully convnet architectures for image processing to video data, capturing well the inherent dynamics without loosing spatial resolution. Some works have proposed to use recurrent neural networks (RNNs) (Siam *et al.*) or their variants, such as long short term memory (LSTM) (Kalchbrenner *et al.*, 2016; Fayyaz *et al.*) or gated recurrent unit (GRU) (Tokmakov *et al.*, 2017b) networks. However, employing RNNs introduces a large number of additional parameters. Consequently, these methods need much more densely annotated video data for training which is quite costly to obtain. Thus, it is highly desirable, to develop architectures that can learn from video volumes without the cost of additional training data or model complexity. Some methods propose to use 3D convolutions to incorporate spatial and temporal information (Varol *et al.*, 2016; Tran *et al.*, 2015). However, it is not clear if temporal dimension can be processed in a similar manner as the spatial, as the presence of scene and camera motion makes association of the pixels difficult. Therefore, the use of fixed-sized spatio-temporal receptive fields may not be the ideal solution, particularly for segmentation task. Overall, encoding temporal information in CNNs for video segmentation still stays an open problem.

Object view changes over time. In pixel-level object tracking the goal is to segment an object in a video, for which the only available piece of information is its segmentation mask in the first frame. For a human this is usually a very simple task and the limited amount of information in the first frame is more than enough to track the object. Changes in appearance and camera viewpoint, shape variations, occlusions or similar looking instances do not pose a significant challenge as humans are able to leverage strong objectness and semantic priors as well as distinguish the unique discriminative parts of the target object. However, for the convnets for each new frame labelling pixels as object/non-object of interest is a challenging task. During training the network can learn a notion of objectness, but during test time it does not know which of the multiple possible objects in the video sequence it should segment. The challenge is then: how to inform the network which instance to segment? The ground truth mask of the first frame can provide to the network the information about the specific appearance of the object of interest. But the appearance can change dramatically over time due to illumination changes or dis-occlusions as well as the object shape, and its original location in the frame can alter. The network might not adapt well to these drastic changes, causing the loss of the object or the

drift of the mask.

1.1.3.2 *Contributions*

Here we discuss the contributions of the thesis for the *learning to track objects in videos via CNNs*.

The first contribution is approaching video object segmentation as guided instance segmentation. In Chapter 9 we propose to use a pixel labelling convnet for frame-by-frame segmentation, utilizing the object mask from the previous frame to enable the temporal context. For each video frame the convnet is guided towards the object of interest by feeding in the previous frame mask estimate as an additional input channel to the network. The extra mask channel is meant to inform the network which instance to segment by providing its approximate location and shape in the current frame. Then the task of the network is to refine the provided rough estimate of the object mask based on the content of the current frame. In this way we only need to consider one RGB frame at a time along with the previous frame binary mask. This allows to avoid using expensive densely annotated video data for training and enables training the convnet with static images only.

The second contribution is online fine-tuning on the given first frame annotation of the test video. Using augmented versions of this single frame groundtruth we fine-tune the model to become more specialized for the specific object instance at hand. The network learns to capture the appearance of the object of interest and to ignore the background. This step has shown to be very effective (see Chapter 9) for pixel-level object tracking, as it allows to easily adapt to new objects and scenes and to mitigate the problem of domain shift.

The third contribution is an integration of motion cues into the pixel-level object tracking convnet. Fusing the appearance and motion cues allows to better exploit the information inherent to video and enables the model to segment well both static and moving objects. For this we propose to employ optical flow magnitude as an additional input channel to the network. When the object is moving relative to background, the flow magnitude provides a very reasonable estimate of the object mask, giving complementary information to the RGB image. We show in Chapter 10 that integrating optical flow provides consistent improvement of the performance across different video object segmentation benchmarks.

The fourth contribution is a novel data synthesis technique which allows to reduce the dependence on large video and image datasets for training the tracking convnet. To ensure a sufficient amount of training samples close to the test video we propose to synthesize training data using the given first frame image and its annotation mask. The aim is to produce a large set of reasonably realistic images which capture plausible variations in future video frames, and thus is, by design, close to the test video. We call this synthesis strategy lucid data dreaming (see Chapter 10). Employing the lucid dream images for training allows to achieve high-quality tracking results without requiring external data (neither images with mask annotations nor annotated videos) and hence to avoid the problem of the domain shift.

The fifth contribution is showing that training a convnet for object tracking can be done with only few annotated frames. In the extreme case, with only a single annotated frame and zero pre-training, competitive tracking results can be obtained. Our experiments in Chapter 10 indicate that increasing the number of training images does not always improve the resulting quality of the tracker and using training samples close to the test domain is more effective than adding more training data from related domains. This changes the mindset regarding how much general objectness knowledge is required to approach pixel-level object tracking task, and more broadly how much annotated data is required to train a convnet depending on the task at hand.

1.2 OUTLINE OF THE THESIS

In this section we summarize each chapter of the thesis. In addition, we also indicate the respective publications and collaborations with other researches.

The first part of the thesis (Chapters 3 – 5) focuses on image segmentation with weaker forms of supervision, while the second (Chapters 6 – 8) and the third (Chapters 9 – 10) parts contributes to video segmentation via graphs or convnets respectively.

Chapter 2: Related work. In this chapter we review the related work on image and video segmentation, as well as other related topics. We analyze the relations of previous and subsequent works to the research presented in this thesis.

Chapter 3: Weakly Supervised Object Boundaries. This chapter presents weakly supervised approach for object boundary detection, which can be employed in image and video segmentation to correctly delineate the object mask. We propose to detect class-specific object boundaries using only bounding box supervision by generating pixel-level approximate groundtruth to train a detector. We show that high-quality object boundaries can be obtained by employing box annotations alone.

The content of this chapter corresponds to the CVPR 2016 publication: "Weakly Supervised Object Boundaries" (Khoreva *et al.*, 2016b), which was accepted as a *Spotlight* (9.7% acceptance rate). Anna Khoreva was the lead author of this paper. Mohamed Omran contributed with the implementation and training of the HED boundary detector.

Chapter 4: Weakly Supervised Instance and Semantic Segmentation. In this chapter we explore learning instance and semantic segmentation from bounding boxes. Starting from box annotations we show how standard computer vision techniques can be used to generate approximate segmentation annotation. We use the learned boundaries from Chapter 3 to improve the object mask estimation from bounding boxes. With the proposed technique we not only outperform competitive weakly supervised methods, but also get close to the performance of methods with full supervision.

The content of this chapter corresponds to the CVPR 2017 publication: "Simple Does It: Weakly Supervised Instance and Semantic Segmentation" (Khoreva *et al.*, 2017a). Anna Khoreva was the lead author of this paper. Jan Hosang contributed with the instance segmentation experiments with DeepMask in Section 4.5.

Chapter 5: Exploiting Saliency for Segmentation from Image Labels. This chapter presents a semantic segmentation approach using only image label supervision. We show how to obtain the full extent of the object by employing a saliency model as an additional source of information.

The content of this chapter corresponds to the CVPR 2017 publication: "Exploiting Saliency for Object Segmentation from Image Level Labels" (Oh *et al.*, 2017). Seong Joon Oh was the lead author of this paper. Anna Khoreva contributed the model for weakly supervised saliency segmentation and corresponding experiments, as well as overall discussion.

Chapter 6: Learning Must-Link Constraints for Video Segmentation. In this chapter we propose how to learn and integrate must-link constraints into graph-based video segmentation. We demonstrate that the integration of learned must-link constraints allows to reduce the computational load for graph partitioning as well as improve the overall video segmentation quality.

The content of this chapter corresponds to the GCPR 2014 publication: "Learning Must-Link Constraints for Video Segmentation Based on Spectral Clustering" (Khoreva *et al.*, 2014). Anna Khoreva was the lead author of this paper.

Chapter 7: Classifier Based Graph Construction. The chapter investigates how to construct a better graph for video segmentation. We propose to learn both the edge topology and weights of the graph by means of a classifier over superpixel features. Addressing the graph construction helps to achieve better performance without altering the graph partitioning or the underlying features.

The content of this chapter corresponds to the CVPR 2015 publication: "Classifier Based Graph Construction for Video Segmentation" (Khoreva *et al.*, 2015). Anna Khoreva was the lead author of this paper.

Chapter 8: Improved Image Boundaries for Video Segmentation. In this chapter we focus on better superpixels for video segmentation. We analyze the existing superpixel methods and show that superpixels extracted from boundaries achieve the best performance. To obtain better superpixels we propose to improve boundary estimation via fusion of image and time domain cues. By employing superpixels generated from better boundaries we observe consistent improvement across different video segmentation approaches, including the method proposed in Chapter 7.

The content of this chapter corresponds to the ECCV 2016 Workshops publication: "Improved Image Boundaries for Better Video Segmentation" (Khoreva *et al.*, 2016a). Anna Khoreva was the lead author of this paper.

Chapter 9: Learning Video Object Segmentation from Static Images. This chapter presents an approach to semi-supervised video object segmentation via convnets. Instead of relying on densely annotated video data we propose to train a video segmentation network using static images only. We treat the problem as a guided instance segmentation and process the video per-frame. The temporal context is enabled by using the guidance from the previous frame mask as an additional input channel to the network. To learn the specific appearance of the target object we fine-tune the model per-video for a small number of iterations.

The content of this chapter corresponds to the CVPR 2017 publication: "Learning Video Object Segmentation from Static Images" (Perazzi *et al.*, 2017), which was accepted as a *Spotlight* (8% acceptance rate). This paper is based on collaboration with Disney Research and ETH Zurich, Switzerland. Anna Khoreva and Federico Perazzi contributed equally to the paper.

Chapter 10: Lucid Data Dreaming for Object Tracking. In this chapter we propose to reduce the dependence on large image (as in Chapter 9) and video pixel-level annotated datasets for training the mask tracking convnet. We introduce a data synthesis method, which creates a large number of training examples from the first annotated frame. This approach allows to reach competitive performance even when training from only a single annotated frame. We demonstrate experimentally that using a larger training set is not automatically better, and that for the tracking task a smaller training set that is closer to the target domain is more effective.

The content of this chapter partially corresponds to the CVPR 2017 Workshops online publication: "Lucid Data Dreaming for Object Tracking" (Khoreva *et al.*, 2017b). This work is based on a collaboration with Google Research and the Computer Vision Group at the University of Freiburg. Anna Khoreva was the lead author of this paper. Eddy Ilg contributed with the FlowNet2.0 model for optical flow estimation. The proposed approach to pixel-level object tracking has taken the second place in the 2017 DAVIS Challenge on Video Object Segmentation.

Chapter 11: Conclusions and future perspectives. In this chapter we summarize the thesis and discuss possible future research directions for image and video segmentation.

RELATED WORK 2

Both image and video segmentation have a long history of research. In this chapter we give an overview of related work, focusing on the directions explored in this thesis, and discuss differences and similarities to the methods proposed in this work.

This chapter is organized as follows. We first present recent work on object boundary detection in Section 2.1, which often is used to improve the quality of image and video segmentation (see Chapters 4 and 6-8). Section 2.2 considers work on semantic image segmentation with different forms of supervision. Section 2.3 discusses recent advances on instance segmentation. Section 2.4 goes into details about work on fully automatic and human-guided video segmentation.

2.1 OBJECT BOUNDARY DETECTION

First we will give an overview over recent work on object boundary detection. In the following we distinguish two types of boundaries: generic and class-specific object boundaries. Generic boundary detection methods aim to detect external and internal edges of "things" and "stuff". However, some perception studies (Kourtzi and Kanwisher, 2001; Hsieh *et al.*, 2010) suggest that humans employ object-level reasoning when judging if a particular pixel is a boundary. Therefore there is a need to detect class-specific object boundaries - external object boundaries of certain semantic classes, which are more consistent with humans reasoning. Both types of boundaries can be used to aid a number of high-level vision tasks, in particular for image (Kirillov *et al.*, 2017; Kokkinos, 2016) and video segmentation (Galasso *et al.*, 2012; Yi and Pavlovic, 2015).

2.1.1 Generic boundaries

Early methods for generic boundary detection are based on a fixed prior model of what constitutes a boundary, e.g. the Sobel detector (Kittler, 1983), zero-crossings (Marr and Hildreth, 1980), and the widely adopted Canny detector (Canny, 1986).

Modern methods leverage machine learning to push performance, enabled by the existence of manually annotated datasets, e.g. BSDS (Arbeláez *et al.*, 2011). A range of techniques have been proposed, from well crafted features and simple classifiers (Statistical Edges (Coughlan *et al.*, 2003), Pb (Martin *et al.*, 2004), gPb (Arbeláez *et al.*, 2011)) to powerful decision trees over fixed features (Sketch Tokens (Lim *et al.*, 2013), SE (Dollár and Zitnick, 2015), OEF (Hallman and Fowlkes, 2015)), and recently to end-to-end learning via Convolutional Neural Networks (CNNs,

convnets) (N^4-Fields (Ganin and Lempitsky, 2014), DeepContour (Shen *et al.*, 2015), DeepEdge (Bertasius *et al.*, 2015a), HFL (Bertasius *et al.*, 2015b), HED (Xie and Tu, 2015)).

Convnets for boundary detection are usually pre-trained on large classification datasets, so as to be initialized with reasonable features, and then trained on boundary datasets (Arbeláez *et al.*, 2011; Hariharan *et al.*, 2011). The more sophisticated the model, the more data is needed to learn it.

N^4-Fields (Ganin and Lempitsky, 2014) rely on dictionary learning and the use of a nearest neighbor algorithm within a CNN framework to predict contours. DeepContour (Shen *et al.*, 2015) learns deep features using shape information. DeepEdge (Bertasius *et al.*, 2015a) and HFL (Bertasius *et al.*, 2015b) use features generated by pre-trained CNNs to regress contours, showing that object-level information provides powerful cues for boundary detection. HED (Xie and Tu, 2015) proposes an end-to-end framework to boost the efficiency and accuracy of contour detection, by combining multi-scale and multi-level visual responses from the intermediate layers of a network. Kokkinos (2016) build upon Xie and Tu (2015), improving the results by a careful design of the loss function, a multi-resolution architecture, additional training data and globalization.

Other than pure boundary detection methods, to improve or to generate closed contours segmentation techniques, such as gPb-owt-ucm (Arbeláez *et al.*, 2011), F&H (Felzenszwalb and Huttenlocher., 2004), MCG (Pont-Tuset *et al.*, 2016), and COB (Maninis *et al.*, 2017), can also be used.

A few works have addressed unsupervised detection of generic boundaries (Isola *et al.*, 2014; Li *et al.*, 2015). PMI (Isola *et al.*, 2014) detects boundaries by modeling them as statistical anomalies amongst all local image patches, reaching competitive performance without the need for training. Li *et al.* (2015) propose to train edge detectors using motion boundaries obtained from a large corpus of video data in place of human supervision. Both approaches reach similar detection performance.

Several works have proposed to make use of the learned boundaries to improve higher-level tasks, such as semantic image labelling (Bertasius *et al.*, 2015b; Kokkinos, 2016; Bertasius *et al.*, 2016; Chen *et al.*, 2016a) or instance segmentation (Kirillov *et al.*, 2017; Hayder *et al.*, 2017). In Chapters 6-8 we employ boundary detection to extract superpixels and to estimate a pairwise affinity for graph-based video segmentation. In Chapter 4 we employ pair-wise terms based on the learned boundaries to improve the estimation of the object mask via GrabCut (Rother *et al.*, 2004).

2.1.2 Class-specific object boundaries

In many applications, there is an interest to focus on boundaries with high-level semantics for specific object classes. The class-specific object boundary detectors are then trained or tuned to the classes of interest. This problem is more recent and relatively unexplored.

Hariharan *et al.* (2011) introduced the SBD dataset to measure for this task over the 20 pascal categories.Hariharan *et al.* (2011) propose to re-weight generic

boundaries using the activation regions of a detector. Uijlings and Ferrari (2015) propose to train class-specific boundary detectors, and weighted them at test time according to an image classifier. Premachandran *et al.* (2017) introduced the PASCAL Boundaries dataset, a class-agnostic semantic boundary dataset with annotations between 459 semantic classes, including both foreground objects and different types of background. To solve the task they propose a multi-scale convnet-based class-agnostic semantic boundary detector.

In Chapter 3 we introduce a method to detect object-specific boundaries using only bounding box supervision, without using expensive boundary annotations. Multiple works have addressed weakly supervised learning for object localization (Oquab *et al.*, 2015; Cao *et al.*, 2015), object detection (Prest *et al.*, 2012; Wang *et al.*, 2014a), or semantic labelling (Vezhnevets *et al.*, 2011; Xu *et al.*, 2015; Pinheiro and Collobert, 2015). To the best of our knowledge there is no previous work attempting to learn object boundaries in a weakly supervised fashion, using just bounding box supervision.

2.2 SEMANTIC SEGMENTATION

Semantic segmentation requires understanding an image at pixel level, where each pixel in the image is assigned to a certain semantic class. Apart from recognizing the object, the goal is also to delineate the boundaries of each object and output dense pixel-wise predictions. The accuracy of semantic segmentation models strongly correlates with the amount of available training data. Collecting and annotating pixel-wise data has become a bottleneck. This problem has raised interest in exploring different means of weaker forms of supervision and investigating what is the minimal supervision needed to reach quality comparable to the fully supervised case.

2.2.1 Fully supervised semantic labelling

Even when pixel-level annotations are provided (fully supervised case), the task of semantic labelling is far from being solved (Everingham *et al.*, 2015; Lin *et al.*, 2014; Hariharan *et al.*, 2011).

Most of the successful semantic labelling methods developed in the previous decade rely on hand-crafted features combined with flat classifiers, such as decision forests (Shotton *et al.*, 2008), boosting (Shotton *et al.*, 2009; Tu and Bai, 2010) or support vector machines (Fulkerson *et al.*, 2009). Major improvements have been achieved by integrating richer information from context (Carreira *et al.*, 2012; George, 2015) and structured prediction techniques (Krähenbühl and Koltun, 2011; Gould *et al.*, 2009; He *et al.*, 2004; Ladicky *et al.*, 2009; Carreira and Sminchisescu, 2012), though the performance of these methods has always been restricted by the limited power of the features and relatively shallow models.

More recent works employ the top convolutional layers of a pre-trained CNN (Krizhevsky *et al.*, 2012; Simonyan and Zisserman, 2015) as feature representations

(Girshick *et al.*, 2014; Caesar *et al.*, 2015; Dai *et al.*, 2015b; Hariharan *et al.*, 2014; Mostajabi *et al.*, 2015; Sharma *et al.*, 2015; Farabet *et al.*, 2013). These features can represent the bounding box around the object (Caesar *et al.*, 2015; Girshick *et al.*, 2014) or respect the object shape (Dai *et al.*, 2015b; Hariharan *et al.*, 2014; Mostajabi *et al.*, 2015; Sharma *et al.*, 2015; Farabet *et al.*, 2013). Another approach using recurrent neural networks (Pinheiro and Collobert, 2014) merges several low resolution predictions to output a full resolution prediction. These techniques are already an improvement over hand-crafted features but their ability to correctly delineate objects is poor.

The fully convolutional networks (FCNs) for semantic image segmentation introduced by Long *et al.* (2015) have proven to be particularly effective for the task and have given rise to a wide range of segmentation research using end-to-end training. Long *et al.* (2015) transformed fully-connected layers of a CNN into convolutional layers, enabling dense pixel-wise classification using CNN architectures that were pre-trained on ImageNet, such as VGG (Simonyan and Zisserman, 2015).

However, the repeated combination of pooling and down-sampling at consecutive layers of the networks originally designed for image classification results in output feature maps at significantly reduced spatial resolution and poorly localized object boundaries. Pooling operations are highly desirable for recognizing objects in images, as it increases the size of the receptive field and makes the network robust against small translations in the image, but when applied to segmentation they significantly decrease localization performance.

To overcome this problem and obtain a pixel-accurate segmentation various strategies have been proposed. Most of them build upon classification architectures such as VGG (Simonyan and Zisserman, 2015) or ResNet (He *et al.*, 2016). Some approaches (Chen *et al.*, 2015; Liu *et al.*, 2015; Long *et al.*, 2015) extract features from intermediate layers via skip-layer connections, allowing information to propagate directly from early, high-resolution layers to deeper layers. To reduce the pooling factor of the pre-trained network, methods of Chen *et al.* (2016b) and Yu and Koltun (2016) use atrous convolution, also known as dilated convolution.

Noh *et al.* (2015) and Hong *et al.* (2015) propose an encoder/decoder network. The encoder computes low-dimensional feature representations via a sequence of pooling and convolution operations. The decoder, stacked on top of the encoder, then learns an upscaling of these low-dimensional features via subsequent unpooling and deconvolution operations (Zeiler *et al.*, 2011). Similarly to Noh *et al.* (2015), Badrinarayanan *et al.* (2015) re-use the pooling indices from the encoder and learn extra convolutional layers to densify the feature responses. Ronneberger *et al.* (2015) add skip connections from the encoder features to the corresponding decoder activations.

An alternative approach is to apply post-processing smoothing operations to the output of a CNN in order to obtain more consistent predictions (Krähenbühl and Koltun, 2011; Kolmogorov and Zabih, 2004; Barron and Poole, 2015). Most commonly, conditional random fields (CRF) (Krähenbühl and Koltun, 2011) are applied on the network output to capture long range dependencies between pixels

(Chen *et al.*, 2016b, 2015; Kokkinos, 2016; Lin *et al.*, 2016c). Further improvements have been shown in Schwing and Urtasun (2015) and Zheng *et al.* (2015) by jointly training both the CRF and CNN components. Jampani *et al.* (2016b) propose to learn bilateral convolutions, while Chandra and Kokkinos (2016) and Vemulapalli *et al.* (2016) combine gaussian CRF with CNN. Yu and Koltun (2016) and Liu *et al.* (2017) employ cascade of several extra convolutional layers to gradually capture long range context information.

Several works (Zhao *et al.*, 2016; Chen *et al.*, 2016b; Liu *et al.*, 2015; Ghiasi and Fowlkes, 2016) employ spatial pyramid pooling to capture objects at multiple scales. Spatial pyramid pooling probes an incoming feature map with filters or pooling operations at multiple rates and multiple effective fields-of-views, thus capturing context in different ranges. Ghiasi and Fowlkes (2016) employ multi-scale predictions via a Laplacian pyramid reconstruction network to successively improve the boundary adherence. The image-level features are exploited in the work of Liu *et al.* (2015) for global context information. Zhao *et al.* (2016) perform spatial pooling at several grid scales via a pyramid scene parsing network. Chen *et al.* (2016b) propose atrous spatial pyramid pooling (ASPP), where parallel atrous convolution layers with different rates capture multi-scale information. Most recently, Chen *et al.* (2017a) propose to augment the ASPP module with image-level features encoding global context and further boosting performance.

Lin *et al.* (2016b) propose a multi-path refinement network that exploits all the information available along the down-sampling process to enable high-resolution predictions using long-range residual connections. Pohlen *et al.* (2017) exploit a ResNet-like architecture that combines multi-scale context with pixel-level accuracy by using two processing streams, one that is processed at full resolution and another that performs down-sampling operations.

All these approaches achieve state-of-the-art performance but require expensive large-scale pixel-level annotations for training. To make the training for new object classes more affordable, there is a need to relax the requirement of high-quality pixel-level annotations. In Chapter 4 and Chapter 5 we explore weaker forms of supervision for semantic segmentation. For comparison with previous work, we base our experiments on the popular DeepLab (Chen *et al.*, 2015, 2016b) architecture.

2.2.2 Weakly supervised semantic labelling

In order to keep annotation cost low, recent work has explored different forms of supervision for semantic labelling: image labels (Pathak *et al.*, 2015b; Papandreou *et al.*, 2015; Pinheiro and Collobert, 2015; Wei *et al.*, 2015; Kolesnikov and Lampert, 2016b; Durand *et al.*, 2017), prior meta-information (Pathak *et al.*, 2015a), points/clicks (Bearman *et al.*, 2015), scribbles (Xu *et al.*, 2015; Lin *et al.*, 2016a), bounding boxes (Dai *et al.*, 2015a; Papandreou *et al.*, 2015), class-agnostic segmentation masks (Chaudhry *et al.*, 2017) and masks from other classes (Hong *et al.*, 2016), web-crawled images (Jin *et al.*, 2017; Wei *et al.*, 2015) and videos (Hong *et al.*, 2017; Tokmakov *et al.*, 2016). (Dai *et al.*, 2015a; Papandreou *et al.*, 2015; Hong *et al.*, 2015; Souly *et al.*, 2017) also

consider the case where a fraction of images are fully supervised. Xu *et al.* (2015) propose a framework to handle different types of annotations.

Bounding box supervision. In Chapter 4 we focus on box level annotations for semantic labelling of objects. The closest related work are thus (Dai *et al.*, 2015a; Papandreou *et al.*, 2015). BoxSup (Dai *et al.*, 2015a) uses a recursive training procedure, where the supervision during training the convnet is object segment proposals and the updated network in turn improves the segments used for training. WSSL (Papandreou *et al.*, 2015) employs an expectation-maximisation algorithm with a bias to enable the network to estimate the foreground regions.

Both BoxSup and WSSL propose new ways to train convnets under weak supervision. Similar to our method in Chapter 4, both of the approaches build upon the DeepLab network (Chen *et al.*, 2015). However, there are a few differences in the network architecture. WSSL employs 2 different variants of the DeepLab architecture with small and large receptive field of view (FOV) size. For each experiment WSSL evaluates both architectures and reports the best result obtained. BoxSup uses their own implementation of DeepLab with a small FOV. In our approach we employ the DeepLab architecture with a large FOV.

More importantly and in contrast to BoxSup and WSSL, in Chapter 4 we show that one can reach better results without modifying the training procedure, compared to the fully supervised case, by instead carefully generating input labels for training from the bounding box annotations.

Image label supervision. In Chapter 5 we employ image labels as the main form of supervision for semantic labelling. Initial work approached this problem by adapting multiple-instance learning (Pathak *et al.*, 2015b) and expectation-maximization techniques (Papandreou *et al.*, 2015; Hou *et al.*, 2016) to the semantic labelling case. Without additional priors only poor results are obtained. Using superpixels to inform about the object shape helps (Pinheiro and Collobert, 2015; Xu *et al.*, 2015) and so does using priors on the object size (Pathak *et al.*, 2015a). Kolesnikov and Lampert (2016b) carefully use a CRF (Krähenbühl and Koltun, 2011) to propagate the seeds across the image during training, while Qi *et al.* (2016) exploit segment proposals (Pont-Tuset *et al.*, 2016) for this.

Most methods compared propose each a new procedure to train a semantic labelling convnet. One exception is the work of Shimoda and Yanai (2016) which shows competitive performance by using an improved form of guided back-propagation (Springenberg *et al.*, 2015). Recognizing the ill-posed nature of the problem, Kolesnikov and Lampert (2016a) and Saleh *et al.* (2016) propose to collect user-feedback as additional information to guide the training of a segmentation convnet. Instead of collecting extra cues from human annotators, Hong *et al.* (2017), Jin *et al.* (2017) and Tokmakov *et al.* (2016) propose to take advantage of data available from the web. Jin *et al.* (2017) exploit web images to build a pipeline to automatically generate segmentation masks for each class and then train a network using these masks. Hong *et al.* (2017) and Tokmakov *et al.* (2016) use web videos as an additional source of

training data, since temporal dynamics in video offers rich information to distinguish objects from background and estimate their shapes more accurately. Souly *et al.* (2017) employ Generative Adversarial Networks (GANs) in semi-supervised learning for semantic segmentation to leverage available image-labeled data and additional synthetic data to improve the fully supervised methods.

Employing attention maps has been shown to improve weakly supervised semantic segmentation (Roy and Todorovic, 2017; Wei *et al.*, 2017). Roy and Todorovic (2017) model visual attention maps using the rectified Gaussian distribution, resulting in an improved spatial smoothness of attention maps per object class. Wei *et al.* (2017) propose an adversarial erasing scheme in order to obtain better attention maps which in turn provide better cues for the training.

The closest related work to the method proposed in Chapter 5 are Wei *et al.* (2015) and Chaudhry *et al.* (2017), which also use saliency as a cue to improve weakly supervised semantic segmentation. However, there are a number of differences to the approach proposed in Chapter 5. In contrast to our work, Wei *et al.* (2015) use a curriculum learning to expose the segmentation convnet with simple images (single object category), and later with more complex ones (multiple objects).For saliency they use a manually crafted class-agnostic method, while we use a deep learning based one, which provides better cues. Their training procedure uses $\sim$40k additional images of the classes of interest crawled from the web; we do not use such class-specific external data. Compared to the work of Wei *et al.* (2015) we report significantly better results, showing in better light the potential of saliency as additional information to guide weakly supervised semantic object labelling.

Most recently, Chaudhry *et al.* (2017) have proposed to combine saliency and attention maps to boost performance. They use fully convolutional attention maps to localize the class-specific regions and a hierarchical approach to discover the class-agnostic salient regions to estimate the extent of the object. These two cues are then combined to obtain pixel-level class-specific approximate groundtruth to train a segmentation network. In contrast to the approach proposed in Chapter 5, they use additional supervision in the form of class-agnostic segmentation masks to train a saliency detector and employ a more powerful ResNet architecture (He *et al.*, 2016).

The seminal work of Vezhnevets *et al.* (2011) proposed to use "objectness" maps from bounding boxes to guide the semantic segmentation task. By using bounding boxes, these maps end up being diffuse; in contrast, saliency maps in Chapter 5 provide sharper object boundaries, thus giving better information to guide the semantic labeller.

2.3 INSTANCE SEGMENTATION

In contrast to instance agnostic semantic labelling that groups pixels by object class, instance segmentation groups pixels by object instances. Instance segmentation is a challenging task because it requires the correct detection of all objects in an image while also precisely segmenting each instance.

Many instance segmentation approaches employ object proposals (Pont-Tuset

and Gool, 2015; Hosang *et al.*, 2015). Some methods first rely on detecting individual objects (Girshick *et al.*, 2014; Dai *et al.*, 2016c; Girshick, 2015; Ren *et al.*, 2015), for which a segmentation mask is then produced. Given a bounding box (e.g. selected by a detector), GrabCut (Rother *et al.*, 2004) variants can be used to obtain an instance segmentation, e.g. (Lempitsky *et al.*, 2009; Cheng *et al.*, 2015a; Taniai *et al.*, 2015; Tang *et al.*, 2015; Yu *et al.*, 2015; Xu *et al.*, 2017).

Earlier methods (Dai *et al.*, 2015b; Hariharan *et al.*, 2014, 2015) make use of bottom-up segments (Pont-Tuset *et al.*, 2016; Uijlings *et al.*, 2013; Krähenbühl and Koltun, 2015, 2014). Hariharan *et al.* (2014) employ Fast-RCNN bounding boxes (Girshick, 2015) and builds a multi-stage pipeline to extract CNN features and segment the object. This framework was later improved by the use of Hypercolumn features (Hariharan *et al.*, 2015) and the utilization of a fully convolutional network (FCN) to encode class-specific shape priors (Li *et al.*, 2016a). Arnab and Torr (2016) further reason about multiple object proposals to handle occlusions where single objects are split into multiple disconnected patches.

DeepMask (Pinheiro *et al.*, 2015) and follow-up works (Pinheiro *et al.*, 2016; Dai *et al.*, 2016a) learn to generate segment proposals using deep CNNs, which are then classified by Fast-RCNN (Girshick, 2015) and refined to achieve better segmentation boundaries. Similarly, Dai *et al.* (2016b) propose a complex multiple-stage cascade that predicts instance masks from bounding-box proposals and semantically labels the masks in sequence. Zagoruyko *et al.* (2016) use a modified R-CNN model (Girshick *et al.*, 2014) to propose instance bounding boxes, followed by further refinement to obtain instance level object masks. Ultimately, these approaches suffer from the fact that they predict a binary mask within the bounding box proposals, making the system slower and less accurate.

Li *et al.* (2017c) propose to combine the object detection approach of Dai *et al.* (2016c) and the segment proposals of Dai *et al.* (2016a) for fully convolutional instance segmentation (FCIS), predicting a set of position-sensitive output channels fully convolutionally. These channels simultaneously address object boxes, masks and semantic classes, making the system fast. However, this approach might experience errors and forged edges on overlapping instances. Bai and Urtasun (2017) combine intuitions from the classical watershed transform and deep learning to produce an energy map of the image where object instances are represented as energy basins. This method has constant runtime regardless of the number of object instances.

Most recently, Mask-RCNN (He *et al.*, 2017) extends Faster-RCNN (Ren *et al.*, 2015) by adding a branch for predicting segmentation masks on each Region of Interest (RoI) in parallel with the existing branch for classification and bounding box regression. The mask branch is a small FCN applied to each RoI, predicting an object mask in a pixel-to-pixel manner. The parallel prediction makes the system simpler and more flexible.

In Chapter 4 we explore weakly supervised training of an instance segmentation convnet. To the best of our knowledge there is no previous work on predicting object masks in a weakly supervised fashion. We use DeepMask (Pinheiro *et al.*, 2015) as a reference implementation for this task. In addition we re-purpose the DeepLabv2

network (Chen *et al.*, 2016b), originally designed for semantic segmentation, for the instance segmentation task.

2.4 VIDEO SEGMENTATION

By partitioning video volume into groups of objects or regions which are coherent in appearance and motion, video segmentation delivers the first step to interpret the video content and thus has shown to be helpful in diverse computer vision tasks, such as semantic video segmentation (as pre-segmentation) (Dai *et al.*, 2015b), activity recognition (by computing features on voxels) (Taralova *et al.*, 2014), or scene understanding (Jain *et al.*, 2013).

In recent years, video segmentation has received significant attention, with great progress on fully automatic methods (Ochs *et al.*, 2014; Yi and Pavlovic, 2015; Xiao and Lee, 2016; Jain *et al.*, 2017; Tokmakov *et al.*, 2017b), human-guided mask propagation techniques (Tsai *et al.*, 2016; Maerki *et al.*, 2016; Nagaraja *et al.*, 2015; Caelles *et al.*, 2017b), and interactive methods (Jain and Grauman, 2016; Spina and Falcão, 2016; Wang *et al.*, 2014b).

2.4.1 Fully automatic methods

Fully automatic or unsupervised video segmentation methods assume no human input on the video during test time. A variety of techniques have been proposed for automatic video segmentation in the past decade. They can be grouped into several categories.

Graph partitioning methods. Video segmentation can be approached as a clustering or graph partitioning problem, under various choices for basic data units: pixels, superpixels/supervoxels, and point trajectories.

The use of graphs is long established in video segmentation (Grundmann *et al.*, 2010; Fragkiadaki and Shi, 2012; Ochs *et al.*, 2014; Galasso *et al.*, 2014). Graph-based video segmentation techniques consist of three main steps: 1. feature computation among pairs of pixels/superpixels/point trajectories; 2. graph construction according to the spatio-temporal neighborhood of the pixels/superpixels or long-term trajectories and edge weight estimation based on the computed features; 3. partitioning of the graph with spatio-temporal clustering. Previous work has used a variety of features (Brox and Malik, 2010; Galasso *et al.*, 2012; Palou and Salembier, 2013), proposed various graph partitioning algorithms (Brox and Malik, 2010; Grundmann *et al.*, 2010; Yi and Pavlovic, 2015), focused on the unary and pairwise terms of the graph (Galasso *et al.*, 2014) and addressed the graph construction (Ren and Malik, 2003; Turaga *et al.*, 2009; Khoreva *et al.*, 2015) itself.

Grundmann *et al.* (2010) employ a greedy agglomerative clustering algorithm that merges two adjacent superpixels if their color difference is smaller than the internal variance of each superpixel. Granularity of the segmentation is controlled

by adding a parameter to the internal variance. However, they only focus on color information and do not make use of spatio-temporal structure. A streaming version of Grundmann *et al.* (2010) was introduced in Xu *et al.* (2012), which provides similar performance, but at a fraction of the cost by using overlapping temporal windows of the video to optimize the segmentation.

Galasso *et al.* (2013) do greedy matching of superpixels by propagating labels from the source frame over time via optical flow. This "simple" method obtains competitive performance on the VSB100 benchmark (Galasso *et al.*, 2013). However, the quality of propagated labels typically decays due to flow estimation errors as the distance from the source frame increases. Another limitation is that this segmentation propagation approach cannot introduce new objects as the label set is fixed based on the source frame and does not contain a label corresponding to a new object.

Robust temporal structure can be extracted by long term motion cues in the form of dense trajectories which are derived using dense optical flow in order to get the final pixel-level segmentation (Lezama *et al.*, 2011; Brox and Malik, 2010; Ochs *et al.*, 2014; Fragkiadaki and Shi, 2012; Sundaram *et al.*, 2010). These methods analyze motion over longer periods, as such long term analysis is able to decrease the intra-object variance of motion relative to the inter-object variance and propagate motion information to frames in which the object remains static. For this, Lezama *et al.* (2011) grouped pixels with coherent motion computed via long-range motion vectors from the past and future frames. Similarly, the work of Brox and Malik (2010) offers a framework for trajectory-based video segmentation through building an affinity matrix between pairs of trajectories. In Fragkiadaki and Shi (2012) discontinuities of embedding density between spatially neighboring trajectories were detected. Ochs and Brox (2012) and Elhamifar and Vidal (2009) propose to incorporate higher order motion models. Most of these techniques employ the spectral clustering paradigm to generate segmentations, while Keuper *et al.* (2015) and Keuper (2017) have shown the advantages of casting the motion trajectory segmentation as a minimum cost multicut problem.

In general, these methods assume homogeneity of motion over the entire object and therefore experience difficulties when different parts of an object exhibit non-homogeneous motion patterns. This problem is amplified with the absence of a strong object prior. Moreover, these approaches require careful selection of a suitable model especially for the trajectory clustering process, which often comes with high computation complexity (Lee *et al.*, 2011). Nevertheless, the long trajectories offer a good cue for inferring long range temporal structure in a video.

Galasso *et al.* (2012) aggregate a set of pairwise affinities in color, optical flow, long trajectory correspondences and adjacent object boundary. With aggregated pairwise affinities, spectral clustering is adopted to infer segment labels. Spectral clustering is one of the standard graph partitioning techniques for video segmentation.

Spectral methods, stemming from the seminal work of Shi and Malik (2000) and Ng *et al.* (2001), have received much attention from the theoretical viewpoint (von Luxburg, 2007; Hein and Bühler, 2010), and proven to be successful for video segmentation (Fragkiadaki and Shi, 2012; Ochs *et al.*, 2014; Khoreva *et al.*, 2015; Keuper

and Brox, 2016; Galasso *et al.*, 2014). Spectral clustering is suitable due to its ability to include long-range affinities (Galasso *et al.*, 2012; Sundaram and Keutzer, 2011) and its global view on the problem (Fowlkes and Malik, 2004), providing balanced solutions. However, one of the important limitations of spectral methods is its large computational demand. The large demands of spectral techniques (Sundaram and Keutzer, 2011; Galasso *et al.*, 2012) are particularly clear in the case of high-quality video datasets, such as VSB100 (Galasso *et al.*, 2013), limiting their current large-scale applicability.

In Chapter 6, we propose to integrate must-link constraints to overcome this limitation. This allows to reduce the size of the problem, while preserving the original optimization objective for all partitions satisfying the must-link constraints. Galasso *et al.* (2014) propose another way to reduce the size of the graph by incorporating a reweighing scheme, such that the resulting segmentation is equivalent to that of the full graph. The equivalence is considered in terms of the normalized cut and of its spectral clustering relaxation. This graph reduction allows reducing runtime and memory consumption.

While graph reduction techniques have been explored and various graph partition algorithms have been proposed, surprisingly little attention has been devoted on how to construct a graph to obtain the best video segmentation performance. In Chapter 7 we argue that constructing the underlying graph is a crucial step for best performance of graph-based methods and focus on learning the graph topology as well as the edge weights. By learning the graph, without changes to the graph partitioning method, we improve the results of Galasso *et al.* (2014), while significantly reducing its runtime, as the learnt graph is much sparser.

While most of the above methods employ superpixels, to the best of our knowledge, none of them examines the quality of the respective superpixels for graph-based video segmentation. The graph nodes - superpixels - are the starting point for unary and pairwise terms, and thus directly impact the final segmentation quality. In Chapter 8 we propose to thoroughly analyze and advance superpixel methods in the context of video segmentation. We show that superpixels extracted from boundaries perform best, and that boundary estimation can be significantly improved via appearance and motion cues. With superpixels generated from better boundaries we observe consistent improvement for different video segmentation methods (Galasso *et al.*, 2013, 2014; Khoreva *et al.*, 2015).

The main limitation of these clustering approaches is their lack of an explicit notion of object appearance. With only low-level bottom-up information they tend to over-segment videos. While this can be a useful intermediate step for some recognition tasks in video, the extracted segments might not directly correspond to objects, making it non-trivial to obtain video object segmentation from this intermediate result.

Object proposal methods. These group of methods focus on generating accurate spatio-temporal tubes of binary masks which are well aligned around the boundaries of the object appearing in the video (Wu *et al.*, 2015; Jain *et al.*, 2014; Fragkiadaki

et al., 2015; Oneata *et al.*, 2014; Banica *et al.*, 2013; Li *et al.*, 2013; Xiao and Lee, 2016).

Banica *et al.* (2013) compute multiple segment proposals per frame and link them across frames using appearance similarity. The method in Li *et al.* (2013) iteratively refine a model to track an object over the video, where the model is initialized with regions from the first frame and hence can experience drifting. Oneata *et al.* (2014) produce multiple video segments by deleting image boundaries that do not exhibit high flow strength and is upper-bounded by the static boundary detector.

Jain *et al.* (2014) developed an extension of the image segment proposal method of Uijlings *et al.* (2013) to videos to obtain object proposals. They compute spatio-temporal proposal regions from an independent motion evidence map, which estimates for each pixel in each frame the likelihood that its motion is different from the dominant motion. While this approach is effective to segment objects that are in motion with respect to the background, it does not provide a mechanism to recover objects that are static in the scene.

Fragkiadaki *et al.* (2015) propose to segment moving objects in videos by ranking spatio-temporal segment proposals according to "moving objectness", i.e. how likely they are to contain a moving object. They generate a set of region proposals in each frame using multiple segmentations on optical flow and static boundaries and filter them by rejecting segments on static background or obvious under- or over-segmentations. The filtered proposals are extended to spatio-temporal tubes using dense point trajectories to recover static segments. Then the tubes are ranked using a moving objectness detector.

In the work of Wu *et al.* (2015) image segment proposals are generated and tracked using learned appearance models. Forward tracking and backtracking schemes are used to track segments starting from every frame and through complete occlusions. Xiao and Lee (2016) first generate a set of spatio-temporal bounding box proposals, and then a space-time GrabCut approach is used to generate per frame segments. They first discover an object's easy instances, and then gradually detect harder instances in temporally-adjacent frames. This allows adaptation to the object's changing appearance over time.

Several methods (Faktor and Irani, 2014; Wang *et al.*, 2015b; Ma and Latecki, 2012; Papazoglou and Ferrari, 2013; Zhang *et al.*, 2013) focused on producing a pixel-wise segmentation of the dominant object in video, both in appearance and motion. These methods first estimate a segment (Papazoglou and Ferrari, 2013; Wang *et al.*, 2015b) or segments (Lee *et al.*, 2011; Zhang *et al.*, 2013), which potentially correspond(s) to the foreground object, and then learn foreground/background appearance models. The learned models are then integrated with other cues, e.g., saliency maps (Wang *et al.*, 2015b), pairwise constraints (Papazoglou and Ferrari, 2013; Zhang *et al.*, 2013), object shape estimates (Lee *et al.*, 2011), to compute the final object segmentation. Alternatives approaches have used long-range interactions between distinct parts of the video to overcome noisy initializations in low-quality videos (Faktor and Irani, 2014), and occluder/occluded relations to obtain a layered segmentation (Taylor *et al.*, 2015).

The main drawbacks of the proposal based techniques are their high computa-

tional cost associated with proposal generation and complicated object inference schemes.

Convnet-based approaches. Recently, convnet-based approaches (Tokmakov *et al.*, 2017a; Jain *et al.*, 2017; Tokmakov *et al.*, 2017b) have become the state of the art for unsupervised video object segmentation. These methods usually process videos per-frame, cast video object segmentation as a binary classification problem (foreground/background) and build the convnet architecture upon the semantic labelling networks (Long *et al.*, 2015; Bansal *et al.*, 2017; Ronneberger *et al.*, 2015).

Tokmakov *et al.* (2017a) propose to learn moving objects in videos via convnets. Their encoder-decoder style network first learns a coarse representation of the optical flow field features, and then refines it iteratively to produce motion labels at the original high-resolution. The output labelling is further refined with an objectness map and CRF (Krähenbühl and Koltun, 2011) to account for errors in optical flow.

Jain *et al.* (2017) propose a framework for segmenting generic objects in videos. They employ a two-stream convnet where individual streams encode generic appearance and motion cues derived from an RGB video frame and its corresponding optical flow. These cues are fused in the network to produce a final object versus background pixel-level segmentation for each video frame. The proposed network can segment both static and moving objects.

Tokmakov *et al.* (2017b) present a two-stream network with a visual memory module. The memory module is a convolutional gated recurrent unit (GRU) that encodes the evolution of the object in the input video sequence. The representation used in the memory module is extracted from two streams — the appearance stream (Chen *et al.*, 2015) which describes static features of objects in the video, and the motion stream (Tokmakov *et al.*, 2017a) which captures motion cues. With these CNN features the GRU component is updated at each frame to learn a visual memory representation of the object in the scene.

Because these methods ignore the first frame annotation and try to segment the most salient object, both in motion and appearance, they fail to distinguish similar looking instances in videos where multiple salient objects move, e.g. flock of penguins.

2.4.2 Human-guided methods

Human-guided video segmentation methods accept human input in the first frame or a subset of frames, then propagate the information to the remaining frames (Nagaraja *et al.*, 2015; Tsai *et al.*, 2016; Badrinarayanan *et al.*, 2010; Jain and Grauman, 2014; Wen *et al.*, 2015; Perazzi *et al.*, 2015; Maerki *et al.*, 2016; Tsai *et al.*, 2010).

Mask propagation techniques. Among this group are semi-supervised or semi-automatic approaches, which assume an object mask in the first frame is known, and the objective is to track the object mask throughout the video. Appearance similarity and motion smoothness across time is used to propagate the first frame

annotation across the video (Maerki *et al.*, 2016; Wang and Shen, 2017; Tsai *et al.*, 2016). These methods usually leverage optical flow and long term trajectories. Existing approaches focus on propagating superpixels (Wen *et al.*, 2015; Jain and Grauman, 2014), constructing graphical models (Maerki *et al.*, 2016; Tsai *et al.*, 2016) or utilizing object proposals (Perazzi *et al.*, 2015).

The label propagation method of (Badrinarayanan *et al.*, 2010) jointly models appearance and semantic information. The key idea is to influence the learning of frame to frame patch correlations as a function of both appearance and class labels. This method was extended to include correlations between non-successive frames using a decision forest classifier by Budvytis *et al.* (2011). Tsai *et al.* (2010) propose to jointly optimize for temporal motion and semantic labels in an energy minimization framework. A sliding window approach is used to process overlapping n-frame grids for efficiency reasons. The result of one n-frame grid is employed as a hard constraint in the next grid and so on.

Fathi *et al.* (2011) use active learning for video segmentation. Each unlabelled pixel is provided a confidence measure based on its distance to a labelled point, computed on a neighbourhood graph. These confidences are used to recommend frames in which more interaction is desired. In the work of Nagaraja *et al.* (2015) video object segmentation is formulated as a spatio-temporal markov random field optimization problem, with a cost function including user input, motion and appearance cues, and spatio-temporal consistency.

Tsai *et al.* (2016) build a graph over pixels and superpixels, uses convnet based appearance terms, and interleave video segmentation with optical flow estimation. For the segmentation model, they construct a multi-level graphical model that consists of pixels and superpixels, each of which plays different roles for segmentation. At the superpixel level, each superpixel is likely to contain pixels from the foreground and background as the object boundary may not be clear. At the pixel level, each pixel is less informative although it can be used for more accurate estimation of motion and segmentation. With the combination of these two levels, the object boundary can be better identified by exploiting both statistics contained in superpixels and details in the pixel level.

Wen *et al.* (2015) construct a graph over neighboring frames connecting superpixels and (generic) object parts to solve the video labeling task. Perazzi *et al.* (2015) propose to build a global graph structure over object proposal segments, and then infer a consistent segmentation. A limitation of methods utilizing long-range connections is that they have to operate on larger image regions such as superpixels or object proposals for acceptable speed and memory usage, compromising on their ability to handle fine details. In contrast, the systems introduced in Chapter 9 and Chapter 10 are efficient at test time due to its feed-forward architecture, operate on a pixel level and generate high quality results in a single pass over the video, without the need for considering more than one frame at a time.

Instead of using superpixels or proposals, Maerki *et al.* (2016) formulate a fully-connected pixel-level graph between frames and efficiently infer the labeling over the vertices of a spatio-temporal bilateral grid (Chen *et al.*, 2007). Because this method

propagates information only across neighboring frames it has difficulties ensuring globally consistent segmentation. On the contrary, our approaches in Chapters 9 and 10 learn the specific appearance of the object of interest via online tuning and therefore produce temporally consistent results.

Box tracking. Classic work on video object tracking focused on bounding box tracking. Many of the insights from these works have been re-used for mask tracking. Some previous works have investigated approaches that improve segmentation quality by leveraging box-level tracking and vice versa (Ren and Malik, 2007; Godec *et al.*, 2011; Duffner and Garcia, 2013; Chockalingam *et al.*, 2009).

Traditional box tracking smoothly updates across time a linear model over hand-crafted features (Henriques *et al.*, 2012; Breitenstein *et al.*, 2009; Kristan *et al.*, 2014). Since then, convnets have been used as improved features (Danelljan *et al.*, 2016, 2015; Ma *et al.*, 2015; Wang *et al.*, 2015a), and eventually to drive the tracking itself (Held *et al.*, 2016; Bertinetto *et al.*, 2016; Tao *et al.*, 2016; Nam *et al.*, 2016; Nam and Han, 2016). Convnet-based approaches need data for pre-training and learning the task.

In Chapter 9 we propose a mask tracking method, which is closely related to convnet-based box trackers of Held *et al.* (2016) and Nam and Han (2016). Held *et al.* (2016) propose to train offline a convnet so as to directly regress the bounding box in the current frame based on the object position and appearance in the previous frame. Nam and Han (2016) propose to use online fine-tuning of a convnet to model the object appearance. Our training strategy in Chapter 9 is inspired by Held *et al.* (2016) for the offline part, and Nam and Han (2016) for the online stage. Compared to the aforementioned methods our approach operates at pixel level masks instead of boxes. Differently from Nam and Han (2016), we do not replace the domain-specific layers, instead fine-tuning all the layers on the available annotations for each individual video sequence.

Convnet-based mask tracking. Following the trend in box-level tracking, recently convnets have been proposed for mask tracking. What makes convnets particularly suitable for the task, is that they can learn what are the common statistics of appearance and motion patterns of objects, as well as what makes them distinctive from the background, and exploit this knowledge when tracking a single particular object. This aspect gives convnets an edge over traditional techniques based on low-level features.

Caelles *et al.* (2017b) train a generic object saliency network, and fine-tune it per-video using the first frame annotation to make the output sensitive to the specific object instance being tracked. The resulting fine-tuned network is then applied on each frame of the video individually. Differently from our approach in Chapters 9 and 10 their segmentation is not guided, and therefore it cannot distinguish multiple instances of the same object. Instead, they incorporate the notion of the object to be segmented based solely on the first frame annotation, which might result in performance decay over time, as the object appearance diverges from the initial

frame. Furthermore, it relies on expensive dense video annotations for pre-training, while we employ static images.

Caelles *et al.* (2017a) extend the work of Caelles *et al.* (2017b) by incorporating the semantic information of an instance segmentation method into the video object segmentation pipeline. More recently, Voigtlaender and Leibe (2017b) have proposed to integrate an online adaptation mechanism into the pipeline of Caelles *et al.* (2017b). To adapt to the object appearance changes they update the network per-frame based on training examples selected online. In order to avoid drift, training examples are carefully selected by choosing pixels for which the network is very certain that they belong to the object of interest as positive examples, and pixels which are far away from the previous frame mask as negative examples.

Jampani *et al.* (2016a) mix convnets with ideas of bilateral filtering. They introduce a Video Propagation Network (VPN) that propagates information forward through video data. The VPN architecture is composed of two components. The first one is a temporal bilateral network that performs image adaptive spatio-temporal dense filtering. The bilateral network allows to connect densely all pixels from current and previous frames and to propagate associated pixel information to the current frame. This is then followed by a standard spatial CNN on the bilateral network output to re-fine and predict the mask for the present video frame.

To cope with frequent occlusions and appearance variations in dynamic scenes, most recently Li *et al.* (2017b) have proposed to employ an adaptive object re-identification module along with our mask propagation introduced in Chapter 9 to retrieve missing instances. Specifically, when missing instances are re-identified with high confidence, they are assigned with a higher priority to be recovered during the mask propagation process. For each retrieved instance, its frame is taken as the starting point and the mask propagation is applied bi-directionally. Both mask propagation and re-identification modules are iteratively applied to the whole video sequence until no more high confidence instances can be found. Following our work in Chapter 10 they employ a two-stream convnet with a RGB and optical flow magnitude branches for mask propagation. However, they adopt the much deeper ResNet network (He *et al.*, 2016) with atrous spatial pyramid pooling and multi-scale testing (Chen *et al.*, 2017a) to increase the model capacity and the resolution of prediction.

The network architecture employed in Chapter 10 is similar to Caelles *et al.* (2017b) and Jain *et al.* (2017). Other than implementation details, there are two differentiating factors. One, we use a different strategy for training: while other works (Caelles *et al.*, 2017b; Jampani *et al.*, 2016a; Voigtlaender and Leibe, 2017b) all rely on consecutive video training frames and/or use an external image datasets (Voigtlaender and Leibe, 2017b; Perazzi *et al.*, 2017; Li *et al.*, 2017b), our approach focuses on using the first frame annotations provided with each targeted video benchmark without relying on external annotations. Two, our approach exploits optical flow more effectively than these previous methods.

Interactive video segmentation. Applications such as video editing for movie production often require a level of accuracy beyond the current state of the art. Thus several works have also considered video segmentation with variable annotation effort, leveraging a human in the loop to provide guidance or correct errors, e.g. (Jain and Grauman, 2016; Fan *et al.*, 2015; Nagaraja *et al.*, 2015). Several methods employ flexible user inputs, enabling human interaction using clicks (Jain and Grauman, 2016; Spina and Falcão, 2016; Wang *et al.*, 2014b) or strokes (Bai *et al.*, 2009; Zhong *et al.*, 2012; Fan *et al.*, 2015).

Albeit our techniques in Chapters 9 and 10 can be adapted for more flexible inputs, we focus on maximizing quality for the non-interactive case with no-additional hints along the video.

Part I

LEARNING TO SEGMENT IMAGES WITH WEAKER FORMS OF SUPERVISION

Convolutional networks have become the de facto technique for many problems in computer vision. Training convolutional networks for applications such as object boundary detection, semantic labelling or instance segmentation requires expensive dense pixel-wise annotations, and thus significant cost is involved in creating large enough training sets. In order to make the training data more affordable, there is a need to relax the constraint of high-quality pixel-level image annotations. In this part we explore weaker forms of supervision for training the networks, such as image label and bounding box annotations, which are cheaper and easier to obtain.

In Chapter 3 we propose a technique to detect class-specific object boundaries using only box supervision by generating pixel-level approximate groundtruth to train a network. We show that bounding box annotations alone suffice to predict high-quality object boundaries without using any object-specific boundary annotations. Motivated by the achieved results we extend this framework in Chapter 4 to other closely related tasks, such as semantic labelling and instance segmentation. We employ the weakly supervised object boundaries proposed in Chapter 3 to improve object mask estimation for generating segmentation annotations and experiment with recursive training. In Chapter 5 we explore a weaker form of supervision for semantic segmentation and propose to train a convnet with image-level annotations of the present object classes. To obtain the full extent of the object we employ a saliency model as an additional source of information. With these weaker forms of supervision we achieve high-quality results, getting close to the full supervision quality.

WEAKLY SUPERVISED OBJECT BOUNDARIES 3

State-of-the-art learning based boundary detection methods require extensive training data. Since labelling object boundaries is one of the most expensive types of annotations, there is a need to relax the requirement to carefully annotate images to make both the training more affordable and to extend the amount of training data.

In this chapter we propose a technique to generate weakly supervised annotations and show that bounding box annotations alone suffice to reach high-quality object boundaries without using any object-specific boundary annotations.

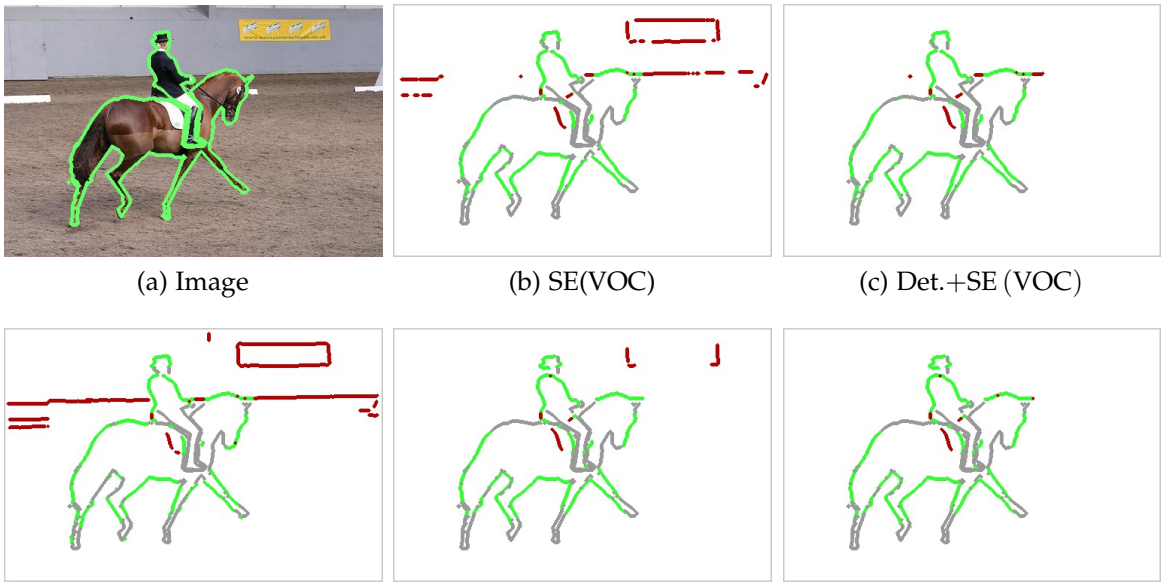
3.1 INTRODUCTION

Boundary detection is a classic computer vision problem. It is an enabling ingredient for many vision tasks such as image/video segmentation (Arbeláez *et al.*, 2011; Galasso *et al.*, 2013), object proposals (Hosang *et al.*, 2015), object detection (Zhu *et al.*, 2015), and semantic labelling (Banica and Sminchisescu, 2015). Rather than image edges, many of these tasks require class specific objects boundaries. These are the external boundaries of object instances belonging to a specific class (or class set).

State-of-the-art boundary detection is obtained via machine learning which requires extensive training data. Yet, instance-wise boundaries are amongst the most expensive types of annotations. Compared to two clicks for a bounding box, delineating an object requires a polygon with 20~100 points, i.e. at least 10$\times$ more effort per object.

In order to make the training of new object classes affordable, and/or to increase the size of the models we train, there is a need to relax the requirement of high-quality image annotations. Hence the starting point of this chapter is the following question: is it possible to obtain object-specific boundaries without having any object boundary annotations at training time?

In this chapter we focus on learning object boundaries in a weakly supervised fashion and show that high quality object boundary detection can be obtained without using any class-specific boundary annotations. We propose several ways of generating object boundary annotations with different levels of supervision, from just using a bounding box oriented object detector to using the boundary detector trained on generic boundaries. For generating weak object boundary annotations we consider different sources, fusing unsupervised image segmentation (Felzenszwalb and Huttenlocher., 2004) and object proposal methods (Uijlings *et al.*, 2013; Pont-Tuset *et al.*, 2016) with object detectors (Girshick, 2015; Ren *et al.*, 2015). We show that bounding box annotations alone suffice to achieve objects boundary estimates



(a) Image (b) SE(VOC) (c) Det.+SE (VOC)

(d) SE(BSDS) (e) SE (weak) (f) Det.+SE (weak)

Figure 3.1: Object-specific boundaries 3.1a differ from generic boundaries (such as the ones detected in 3.1d). The proposed weakly supervised approach drives boundary detection towards the objects of interest. Example results in 3.1e and 3.1f. Red/green indicate false/true positive pixels, grey is missing recall. All methods shown at 50% recall.

with high quality.

We present results using a decision forest (Dollár and Zitnick, 2015) and a convnet edge detector (Xie and Tu, 2015). We report top performance on Pascal object boundary detection (Hariharan *et al.*, 2011; Everingham *et al.*, 2015) with our weak-supervision approaches already surpassing previously reported fully supervised results.

Our main contributions are summarized below:

- We introduce the problem of weakly supervised object-specific boundary detection.
- We show that good performance can be obtained on BSDS, PascalVOC12, and SBD boundary estimation using only weak-supervision (leveraging bounding box detection annotations without the need of instance-wise object boundary annotations).
- We report best known results on PascalVOC12, and SBD datasets. Our weakly supervised results alone improve over the previous fully supervised state of the art.

The rest of this chapter is organized as follows. Section 3.2 describes different types of boundary detection and the considered datasets. In Section 3.3 we investigate the robustness to annotation noise during training. We leverage our findings and

propose several approaches for generating weak boundary annotations in Section 3.4. Sections 3.5-3.8 report results using the two different classifier architectures.

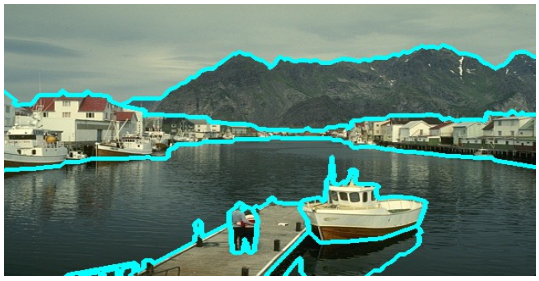
3.2 BOUNDARY DETECTION TASKS

In this work we distinguish three types of boundaries: generic boundaries ("things" and "stuff"), instance-wise boundaries (external object instance boundaries), and class specific boundaries (object instance boundaries of a certain semantic class). For detecting these three types of boundaries we consider different datasets: BSDS500 (Arbeláez *et al.*, 2011; Martin *et al.*, 2001), Pascal VOC12 (Everingham *et al.*, 2015), MS COCO (Lin *et al.*, 2014), and SBD (Hariharan *et al.*, 2011), where each represents boundary annotations of a given boundary type (see Figure 3.2).

BSDS. We first present our results on the Berkeley Segmentation Dataset and Benchmark (BSDS) (Arbeláez *et al.*, 2011; Martin *et al.*, 2001), the most established benchmark for generic boundary detection task. The dataset contains 200 training, 100 validation and 200 test images. Each image has multiple ground truth annotations. For evaluating the quality of estimated boundaries three measures are used: fixed contour threshold (ODS), per-image best threshold (OIS), and average precision (AP). Following the standard approach (Dollár and Zitnick, 2015; Canny, 1986) prior to evaluation we apply a non-maximal suppression technique to boundary probability maps to obtain thinned edges.

VOC. For evaluating instance-wise boundaries we propose to use the PASCAL VOC 2012 (VOC) segmentation dataset (Everingham *et al.*, 2015). The dataset contains 1 464 training and 1 449 validation images, annotated with contours for 20 object classes for all instances. The dataset was originally designed for semantic segmentation. Therefore only object interior pixels are marked and the boundary location is recovered from the segmentation mask. Here we consider only object boundaries without distinguishing the semantics, treating all 20 classes as one. For measuring the quality of predicted boundaries the BSDS evaluation software is used. Following Uijlings and Ferrari (2015) the maxDist (maximum tolerance for edge match) is set to 0.01.

COCO. To show generalization of the proposed method for instance-wise boundary detection we use the MS COCO (COCO) dataset (Lin *et al.*, 2014). The dataset provides semantic segmentation masks for 80 object classes. For our experiments we consider only images that contain the 20 Pascal classes and objects larger than 200 pixels. The subset of COCO that contains Pascal classes consists of 65 813 training and 30 163 validation images. For computational reasons we limit evaluation to 5 000 randomly chosen images of the validation set. The BSDS evaluation software is used (maxDist $= 0.01$). Only object boundaries are evaluated without distinguishing the semantics.



(a) BSDS (Arbeláez *et al.*, 2011)

(b) VOC12 (Everingham *et al.*, 2015)

(c) COCO (Lin *et al.*, 2014)

(d) SBD (Hariharan *et al.*, 2011)

Figure 3.2: Datasets considered.

SBD. We use the Semantic Boundaries Dataset (SBD) (Hariharan *et al.*, 2011) for evaluating class specific object boundaries. The dataset consists of 11 318 images from the trainval set of the PASCAL VOC2011 challenge, divided into 8 498 training and 2 820 test images. This dataset has object instance boundaries with accurate figure/ground masks that are also labeled with one of 20 Pascal VOC classes. The boundary detection accuracy for each class is evaluated using the official evaluation software (Hariharan *et al.*, 2011). During the evaluation process all internal object-specific boundaries are set to zero and the maxDist is set to 0.02. We report the mean ODS F-measure (F), and average precision (AP) across 20 classes.

Note that VOC and SBD datasets have overlap between their train and test sets. When doing experiments across datasets we make sure not to re-use any images included in the test set considered.

Baselines. For our experiments we consider two different types of boundary detectors - SE (Dollár and Zitnick, 2015) and HED (Xie and Tu, 2015) - as baselines. SE is at the core of multiple related methods (SCG, MCG, OEF (Hallman and Fowlkes, 2015)). SE (Dollár and Zitnick, 2015) builds a "structured decision forest" which is a modified decision forest, where the leaf outputs are local boundary patches (16×16 pixels) that are averaged at test time, and the split nodes are built taking into account the local segmentation of the ground truth input patches. It uses binary comparison over hand-crafted edge and self-similarity features as split decisions. By construction this method requires closed contours (i.e. segmentations) as training input. This detector is reasonably fast to train/test and yields good detection quality.

HED (Xie and Tu, 2015) is currently the top performing convnet for BSDS boundaries. It builds upon a VGG16 network pre-trained on ImageNet (Simonyan and Zisserman, 2015), and exploits features from all layers to build its output boundary probability map. By also exploiting the lower layers (which have higher resolution) the output is more detailed, and the fine-tuning is more effective (since all layers are guided directly towards the boundary detection task). To reach top performance, HED is trained using a subset of the annotated BSDS pixels, where all annotators agree (Xie and Tu, 2015). These are so called "consensus" annotations (Hou *et al.*, 2013), and correspond to sparse $\sim$15% of all true positives.

3.3 ROBUSTNESS TO ANNOTATION NOISE

We start by exploring weakly supervised training for generic boundary detection, as considered in BSDS.

Model based approaches such as Canny (Canny, 1986) and F&H (Felzenszwalb and Huttenlocher., 2004) are able to provide low quality boundary detections. We notice that correct boundaries tend to have consistent appearance, while erroneous detections are mostly inconsistent. Robust training methods should be able to pick-up the signal in such noisy detections.

SE. In Figure 3.3 and Table 3.1 we report our results when training a structured decision forest (SE) and a convnet (HED) with noisy boundary annotations. By $(\cdot)$ we denote the data used for training. When training SE using either Canny (SE (Canny)) or F&H (SE (F&H)) we observe a notable jump in boundary detection quality. Comparing SE trained with the BSDS ground truth (fully supervised, SE (BSDS)), with the noisy labels from F&H, SE (F&H) closes up to 80% of the gap between SE (F&H) and SE (BSDS) (ΔAP% column in Table 3.1). Using only noisy weak supervision SE (F&H) is only 3 AP percent points behind from the fully supervised case (76 vs. 79).
We believe that the strong noise robustness of SE can be attributed to the way it builds its leaves. The final output of each leaf is the medoid of all segments reaching it. If the noisy boundaries are randomly spread in the image appearance space, the medoid selection will be robust.

HED. The HED convnet (Xie and Tu, 2015) reaches top quality when trained over consensus annotations. When using all annotations ("non consensus"), its performance is comparable to other convnet alternatives. When trained over F&H the relative improvement is smaller than for the SE case, when combined with SE (denoted "HED(SE (F&H))") it reaches 69 ΔAP% . HED (SE (F&H)) provides better boundaries than SE (F&H) alone, and reaches a quality comparable to the classic gPb method (Arbeláez *et al.*, 2011) (75 vs. 73).

On BSDS the unsupervised PMI methods provides better boundaries than our weakly supervised variants. However, PMI cannot be adapted to provide object-

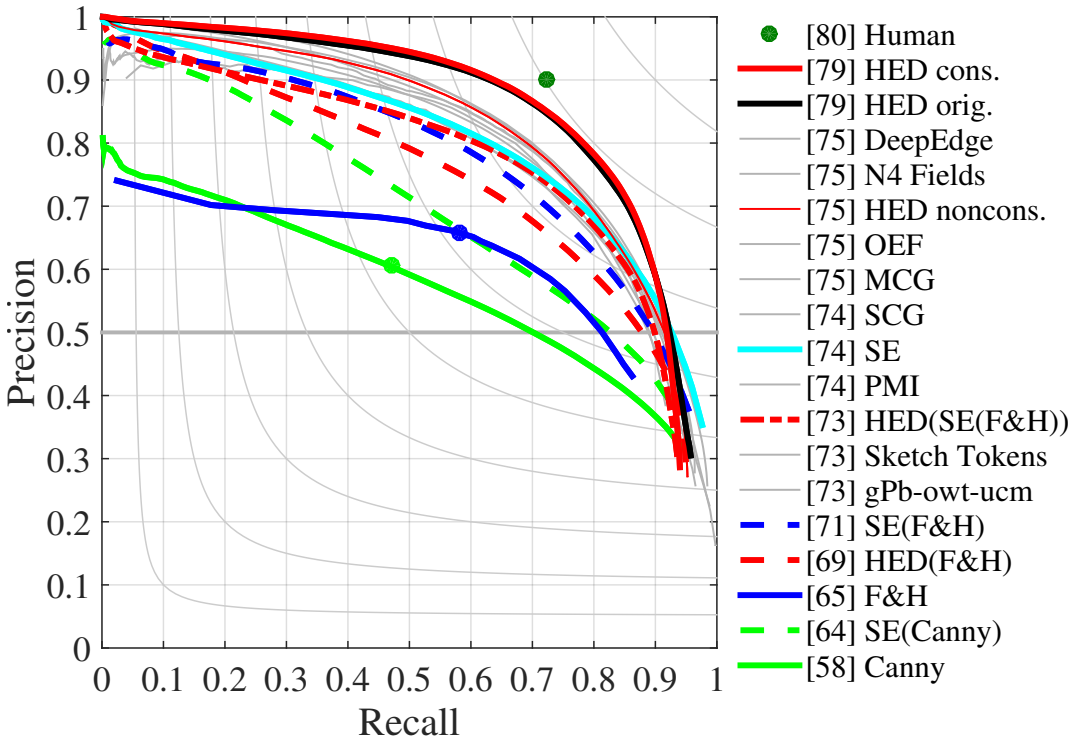


Figure 3.3: BSDS results. Canny and F&H points indicate the boundaries used as noisy annotations. When trained over noisy annotations, both SE and HED provide a large quality improvement.

Family	Method	ODS	OIS	AP	ΔAP%
Unsupervised	Canny (Canny, 1986)	58	62	55	-
	F&H (Felzenszwalb and Huttenlocher., 2004)	64	67	64	-
	PMI (Isola *et al.*, 2014)	74	77	78	-
Trained on ground truth	gPb-owt-ucm (Arbeláez *et al.*, 2011)	73	76	73	-
	SE(BSDS) (Dollár and Zitnick, 2015)	74	76	79	-
	HED(BSDS) noncons. (Xie and Tu, 2015)	75	77	80	-
	HED(BSDS) cons. (Xie and Tu, 2015)	79	81	84	-
Trained on unsupervised boundary estimates	SE (Canny)	64	67	64	38
	SE (F&H)	71	74	**76**	80
	SE (SE (F&H))	72	74	76	80
	SE(PMI)	72	75	77	-
	HED (F&H)	69	72	73	56
	HED (SE (F&H))	73	76	**75**	69

Table 3.1: Detailed BSDS results, see Figure 3.3 and Section 3.3. Underline indicates ground truth baselines, and bold are our best weakly supervised results. $(\cdot)$ denotes the data used for training. ΔAP% indicates the ratio between the same model trained on ground truth, and the noisy input boundaries. The closer to 100%, the lower the drop due to using noisy inputs instead of ground truth.

specific boundaries. For this we need to rely on methods than can be trained, such as SE and HED.

Conclusion. SE is surprisingly robust to annotation noise during training. HED is also robust but to a lesser degree. By using noisy boundaries generated from unsupervised methods, we can reach a performance comparable to the bulk of current methods.

3.4 WEAKLY SUPERVISED BOUNDARY ANNOTATION GENERATION

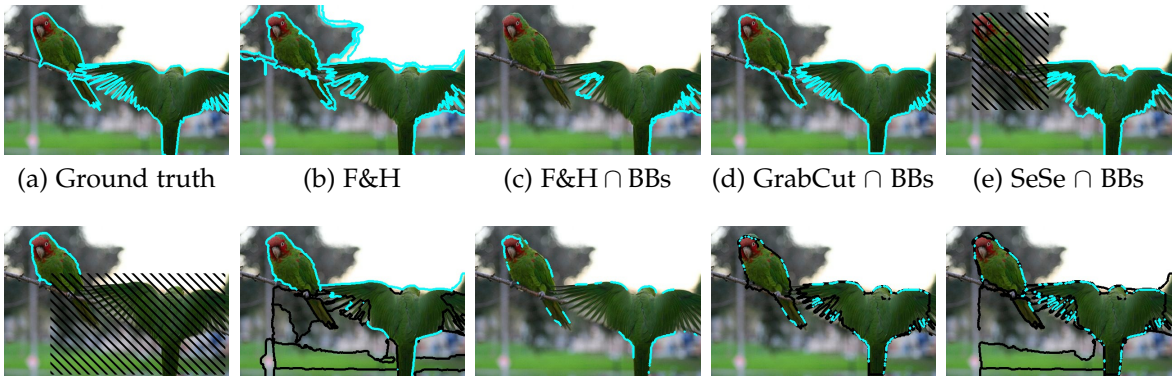
Based on the observations in Section 3.3, we propose to train boundary detectors using data generated from weak annotations. Our weakly supervised models are trained in a regular fashion, but use generated (noisy) training data as input instead of human annotations.

We consider boundary annotations generated with three different levels of supervision: fully unsupervised, using only detection annotations, and using both detection annotations and BSDS boundary annotations (e.g. using generic boundary annotation, but zero object-specific boundaries). In this section we present the different variants of weakly supervised boundary annotations. Some of them are illustrated in Figure 3.4.

BBs. We use the bounding box annotations to train a class-specific object detector (Ren *et al.*, 2015; Girshick, 2015). We then apply this detector over the training set (and possibly a larger set of images), and retain boxes with confidence scores above 0.8. (We also experimented using directly the ground truth annotations, but saw no noticeable difference; thus we report only numbers using the "detections over the training set").

F&H. As a source of unsupervised boundaries we consider the classical graph based image segmentation technique proposed by Felzenszwalb and Huttenlocher. (2004) (F&H). To focus the training data on the classes of interest, we intersect these boundaries with detection bounding boxes from Ren *et al.* (2015) (**F&H** $\cap$ **BBs**). Only the boundaries of segments that are contained inside a bounding box are retained.

GrabCut. Boundaries from F&H will trigger on any kind of boundary, including the internal boundaries of objects. A way to exclude internal object boundaries, is to extract object contours via figure-ground segmentation of the detection bounding box. We use **GrabCut** (Rother *et al.*, 2004) for this purpose. We also experimented with DenseCut (Cheng *et al.*, 2015a) and CNN+GraphCut (Simonyan *et al.*, 2014), but did not obtain any gain; thus we report only GrabCut results.
For the experiments reported below, for **GrabCut** $\cap$ **BBs** a segment is only accepted if a detection from Ren *et al.* (2015) has the intersection-over-union score (IoU) ≥ 0.7. If a detection bounding boxes has no matching segment, the whole region is marked as ignore (see Figure 3.4e) and not used during the training of boundary detectors.



(a) Ground truth (b) F&H (c) F&H ∩ BBs (d) GrabCut ∩ BBs (e) SeSe ∩ BBs

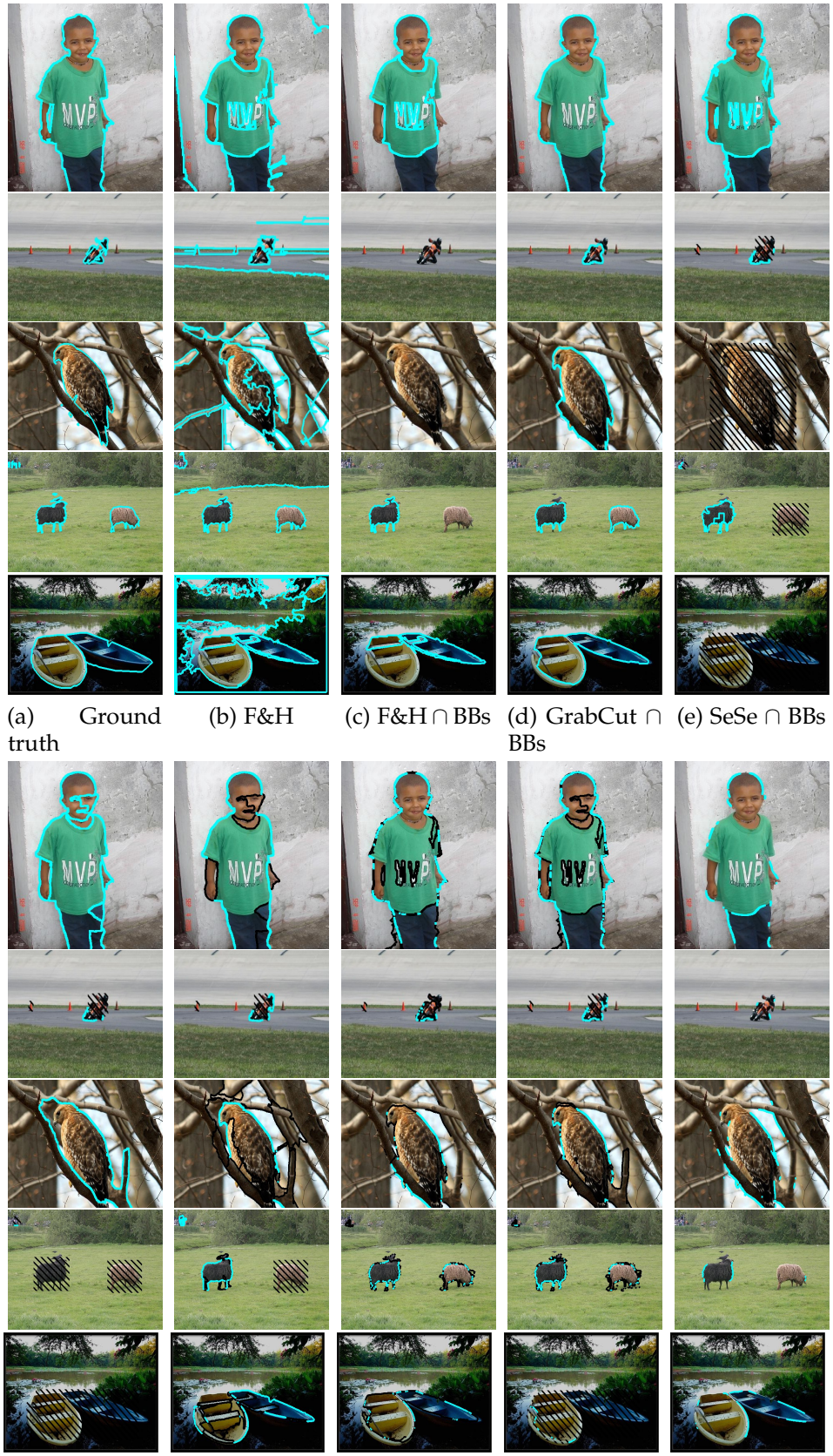
(f) MCG ∩ BBs (g) cons. MCG ∩ BBs (h) SE(SeSe ∩ BBs) (i) cons. S&G∩BBs (j) cons. all methods ∩ BBs

Figure 3.4: Different generated boundary annotations. Cyan/black indicates positive/ignored boundaries.

Object proposals. Another way to bias generation of boundary annotations towards object contours is to consider object proposals. **SeSe** (Uijlings *et al.*, 2013) is based on the F&H (Felzenszwalb and Huttenlocher., 2004) segmentation (thus it is fully unsupervised), while **MCG** (Pont-Tuset *et al.*, 2016) employs boundaries estimated via SE (BSDS) (thus uses generic boundaries annotations).

Similar to GrabCut ∩ BBs, **SeSe ∩ BBs** and **MCG ∩ BBs** are generated by matching proposals to bounding boxes (if IoU $\geq$ 0.9). BBs come from Girshick (2015) with the corresponding object proposals. When more than one proposal is matched to a detection bounding box we use the union of the proposal boundaries as positive annotations. This maximizes the recall of boundaries, and somewhat imitates the multiple human annotators in BSDS. We also experimented using only the highest overlapping proposal, but the union provides marginally better results; thus we report only the later. Since proposals matching a bounding box might have boundaries outside it, we consider them all since the bounding box itself might not cover well the underlying object.

Consensus boundaries. As pointed out in Table 3.1, HED requires consensus boundaries to reach good performance. Thus rather than taking the union between proposal boundaries, we consider using the consensus between object proposal boundaries. The boundary is considered to be present if the agreement is higher than 70%, otherwise the boundary is ignored. We denote such generated annotations as "cons.", e.g. **cons. MCG ∩ BBs** (see Figure 3.4g).
Another way to generate sparse (consensus-like) boundaries, is to threshold the boundary probability map out of an SE (·) model. **SE** (**SeSe ∩ BBs**) uses the top 15% quantile per image as weakly supervised annotations.
Finally, other than consensus between proposals, we can also do consensus between methods. **cons. S&G ∩ BBs** is the intersection between SE (SeSe ∩ BBs), SeSe and GrabCut boundaries (fully unsupervised); while **cons. all methods ∩ BBs** is the



(a) Ground truth (b) F&H (c) F&H ∩ BBs (d) GrabCut ∩ BBs (e) SeSe ∩ BBs

(f) MCG ∩ BBs (g) cons. MCG ∩ BBs (h) cons. S&G ∩ BBs (i) cons. all ∩ BBs (j) SE(SeSe ∩ BBs)

Figure 3.5: Examples of generated boundary annotations. Cyan/black indicates positive/ignored boundaries.

intersection between MCG, SeSe and GrabCut (uses BSDS data).

More examples of generated boundary annotations are in Figure 3.5.

Datasets. Since we generate boundary annotations in a weakly supervised fashion, we are able to generate boundaries over arbitrary image sets. In our experiments we consider SBD, VOC (segmentation), and VOC_+ (VOC plus images from Pascal VOC12 detection task). Methods using $\text{VOC}_+$are denoted using $\cdot_+$ (e.g. $\text{SE}(\text{SeSe}_+ \cap \text{BBs})$).

3.5 STRUCTURED FOREST VOC BOUNDARY DETECTION

In this section we analyse the variants of weakly supervised methods for object boundary detection proposed in Section 3.4 as opposed to the fully supervised ones. From now on we are interested in external boundaries of objects. Therefore we employ the Pascal VOC12, treating all 20 Pascal classes as one. See details of the evaluation protocol in Section 3.2. We start by discussing results using SE; convnet results are presented in Section 3.6.

3.5.1 Training models with ground truth

SE. Figure 3.6 and Table 3.2 show results of SE trained over the ground truth of different datasets (dashed lines). Our results of $\text{SE}(\text{VOC})$ are on par to the ones reported in Uijlings and Ferrari (2015). The gap between $\text{SE}(\text{VOC})$ and $\text{SE}(\text{BSDS})$ reflects the difference between generic boundaries and boundaries specific to the 20 VOC object categories (see also Figure 3.1).

SB. To improve object-specific boundary detection, the situational boundary method SB (Uijlings and Ferrari, 2015), trains 20 class-specific SE models. These models are combined at test time using a convnet image classifier. The original SB results and our re-implementation $\text{SB}(\text{VOC})$ are shown in Figure 3.6. Our version obtains better results (4 percent points gain in AP) due to training the SE models with more samples per image, and using a stronger image classifier (Simonyan and Zisserman, 2015).

Detector + SE. Rather than training and testing with 20 SE models plus an image classifier, we propose to leverage the same training data using a single SE model together with a detector (Girshick, 2015). By computing a per-pixel maximum among all detection bounding boxes and their score, we construct an "objectness map" that we multiply with the boundary probability map from SE. False positive boundaries are thus down-scored, and boundaries in high confidence regions for the detector get boosted. The detector is trained with the same per object boundary annotations used to train the SE model, no additional data is required.

Our $\text{Det.}+\text{SE}(\text{VOC})$ obtains the same detection quality as $\text{SB}(\text{VOC})$ while using

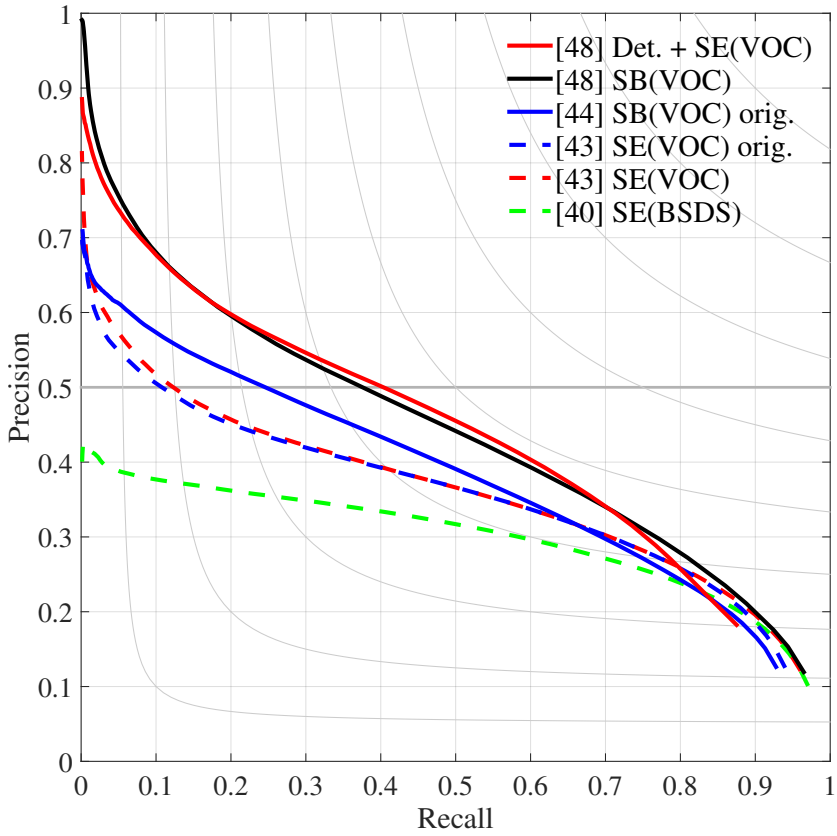


Figure 3.6: VOC12 results, fully supervised SE models. $(\cdot)$ denotes the data used for training. Continuous/dashed line indicates models using/not using a detector at test time. Legend indicates AP numbers.

only a single SE model. These are the best reported results on this task (top of Table 3.2), when using the fully supervised training data.

At the cost of more expensive training and test, one could in principle also combine object detection with the situational boundaries method (Uijlings and Ferrari, 2015), this is out of scope of this thesis and considered as future work.

3.5.2 Training models using weak annotations

Given the reference performance of Det.+SE (VOC), can we reach similar boundary detection quality without using the boundary annotations from VOC?

SE $(\cdot)$. First we consider using a SE model alone at test time. Using only the BSDS annotations leads to rather low performance (see SE (BSDS) in Figure 3.7). PMI shows a similar gap. The same BSDS data can be used to generate MCG object proposals over the VOC training data, and a detector trained on VOC bounding boxes can generate bounding boxes over the same images. We combined them together to generate boundary annotations via MCG $\cap$ BBs, as described in Section

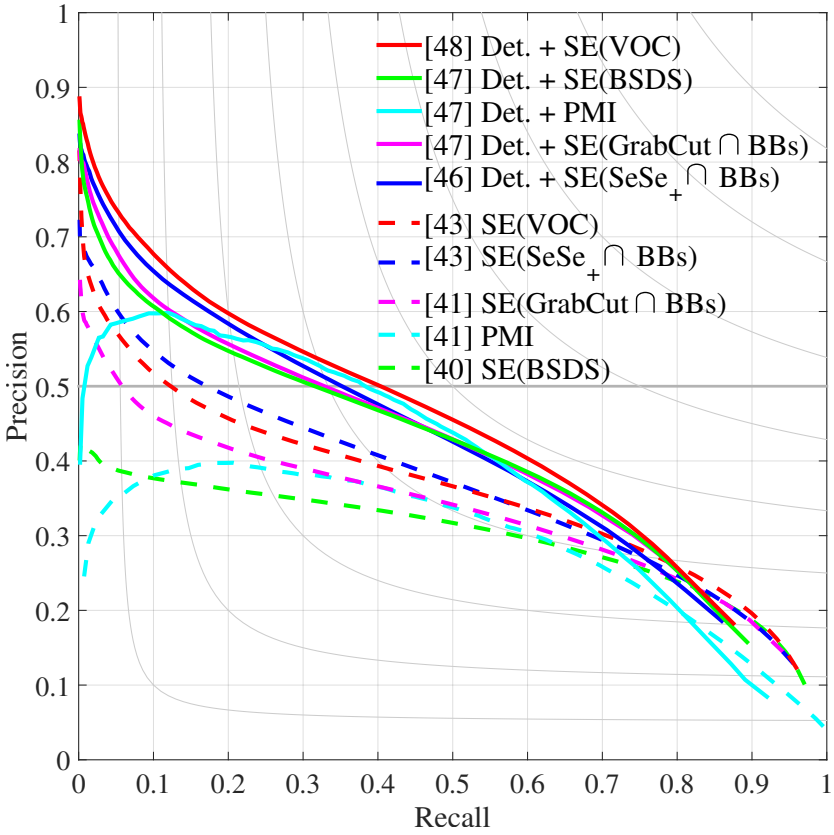


Figure 3.7: VOC12 results, weakly supervised SE models. $(\cdot)$ denotes the data used for training. Continuous/dashed line indicates models using/not using a detector at test time. Legend indicates AP numbers.

3.4. The weak supervision from the bounding boxes can be used to improve the performance of SE (BSDS). By extending the training set to additional pascal images (SE ($\text{MCG}_+ \cap$ BBs) in Table 3.2) we can reach *the same performance* as when using the ground truth VOC data.
We also consider variants that do not leverage the BSDS boundary annotations, such as SeSe and GrabCut. SeSe provides essentially the same result as MCG.

Det.+SE $(\cdot)$**.** Applying object detection at test time squashes the differences among all weakly supervised methods. Det.+PMI shows strong results, but (since not trained on boundaries) fails to reach high precision. The high quality of Det.+BSDS indicates that BSDS annotations, despite being in principle "generic boundaries" in practice reflect well object boundaries, at least in the proximity of an object. This is further confirmed in Section 3.6.
Compared to Det.+BSDS our weakly supervised annotation variants further close the gap to Det.+SE (VOC) (especially in high precision area), even when not using any BSDS data.

Family	Method	Data	Without BBs F	Without BBs AP	Without BBs ΔAP	With BBs F	With BBs AP	With BBs ΔAP
GT	SE	VOC	43	35	-	48	41	-
Other GT	SE	COCO	44	37	2	49	42	1
	SE	BSDS	40	29	-6	47	39	-2
	MCG		41	28	-7	48	39	-2
Weakly supervised	SE	F&H ∩ BBs	40	29	-6	46	36	-5
		GrabCut ∩ BBs	41	32	-3	47	39	-2
		SeSe ∩ BBs	42	35	0	46	39	-2
		$SeSe_+$ ∩ BBs	**43**	**36**	**+1**	46	39	-2
		MCG ∩ BBs	43	34	-1	47	39	-2
		MCG_+ ∩ BBs	43	35	0	**48**	**40**	**-1**
Unsupervised	F&H	-	34	15	-20	41	25	-16
	PMI		41	29	-6	47	38	-3

Table 3.2: VOC results for SE models, see Figures 3.6 and 3.7. Underline indicates ground truth baselines, and bold are our best weakly supervised results.

Conclusion. Based only on bonding box annotations, our weakly supervised boundary annotations enable the Det.+SE model to match the fully supervised case, improving over the best reported results on the task. We also observe that BSDS data allows to train models that describe well object boundaries.

3.6 CONVNET VOC BOUNDARY DETECTION RESULTS

This section analyses the performance of the HED (Xie and Tu, 2015) trained with the weakly supervised variants proposed in Section 3.4. We use our HED re-implementation of HED which is on par performance with the original (see Figure 3.3). We use the same evaluation setup as in the previous section. Figure 3.8 and Table 3.3 show the results.

HED $(\cdot)$**.** The HED(VOC) model outperforms the SE(VOC) model by a large margin. We observe in the test images that HED manages to suppress well the internal object boundaries, while SE fails to do so due to its more local nature.

Even though trained on the generic boundaries HED(BSDS) achieves high performance on the object boundary detection task. HED(BSDS) is trained on the "consensus" annotations and they are closer to object-like boundaries as the fraction of annotators agreeing on the presence of external object boundaries is much higher than for non-object or internal object boundaries.

For training HED, in contrast to SE model, we do not need closed contours and can use the consensus between different weak annotation variants. This results in better performance. Using the consensus between boundaries of MCG proposals

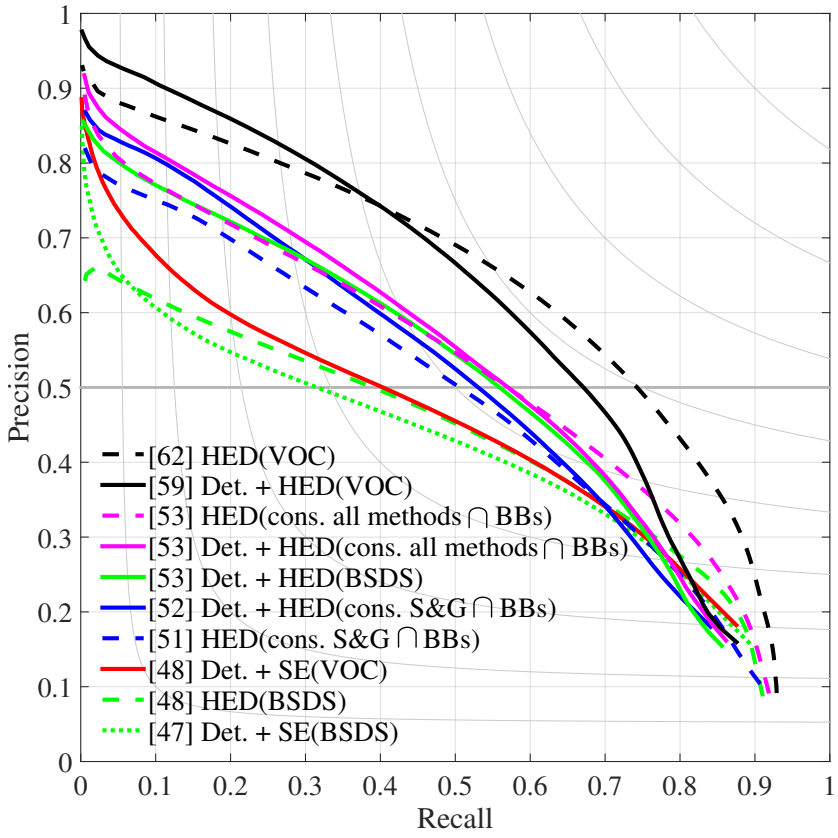


Figure 3.8: VOC12 HED results. $(\cdot)$ denotes the data used for training. Continuous/dashed line indicates models using/not using a detector at test time. Legend indicates AP numbers.

Family	Method	Data	Without BBs			With BBs		
			F	AP	ΔAP	F	AP	ΔAP
GT	SE	VOC	43	35	-	48	41	-
	HED		62	61	26	59	58	17
Other GT	HED	BSDS	48	41	6	53	48	7
		COCO	59	60	25	56	55	14
Weakly supervised	SE	MCG ∩ BBs	43	34	-1	47	39	-2
	HED	SE(SeSe ∩ BBs)	45	37	3	49	40	-1
		MCG ∩ BBs	50	44	9	48	42	1
		cons. S&G ∩ BBs	**51**	**46**	**+11**	**52**	**47**	**+8**
		cons. MCG ∩ BBs	53	50	15	52	49	8
		cons. all methods∩BBs	**53**	**50**	**+15**	**53**	**50**	**+9**

Table 3.3: VOC results for HED models, see Figure 3.8. Underline indicates ground truth baselines, and bold are our best weakly supervised results.

Method	Family	Data	Without BBs			With BBs		
			F	AP	ΔAP	F	AP	ΔAP
SE	GT	COCO	<u>40</u>	<u>32</u>	-	<u>45</u>	<u>37</u>	-
	Other GT	BSDS	34	23	-9	43	33	-4
	Weakly supervised	$SeSe_+ \cap BBs$	**40**	**31**	**-1**	**44**	**35**	**-2**
		$MCG_+ \cap BBs$	39	30	-2	44	35	-2
HED	GT	COCO	60	59	27	56	55	18
	Other GT	BSDS	44	34	2	49	42	5
	Weakly supervised	cons. S&G∩BBs	47	39	7	48	42	5
		cons. all methods∩BBs	**49**	**43**	**+11**	**50**	**44**	**+7**

Table 3.4: COCO results. Underline indicates ground truth baselines.

HED(cons. MCG ∩ BBs) improves AP by 6% compared to using the union of object proposals HED(MCG ∩ BBs) (see Table 3.3).

The HED models trained with weak annotations outperform the fully supervised SE(VOC) and do not reach the performance of HED(VOC). As has been shown in Section 3.3 the HED detector is less robust to noise than SE.

Det.+HED $(\cdot)$**.** Combining an object detector with HED(VOC) (see Det.+HED (VOC) in Figure 3.8) is not beneficial to the performance as the HED detector already has notion of objects and their location due to pixel-to-pixel end-to-end learning of the network.

For HED models trained with the weakly supervised variants, employing an object detector at test time brings only a slight improvement of the performance in the high precision area. The reason for this is that we already use information from the bounding box detector to generate the annotation and the convnet method is able to learn it during training.

Det.+HED (MCG ∩ BBs) outperforms Det.+HED (BSDS) (see Table 3.3). Note that the HED trained with the proposed annotations, generated without using boundary ground truth, performs on par with the HED model trained on generic boundaries (Det.+HED (cons. S&G∩BBs) and Det.+HED (BSDS)in Figure 3.8).

The qualitative results are presented in Figure 3.9 and support the quantitative evaluation.

Conclusion. Similar to other computer vision tasks deep convnet methods show superior performance. Due to the pixel-to-pixel training and global view of the image the convnet models have a notion of object and its location which allows to omit the use of the detector at test time. With our weakly supervised boundary annotations we can gain fair performance without using any instance-wise object boundary or generic boundary annotations and leave out object detection at test time by feeding object bounding box information during training.

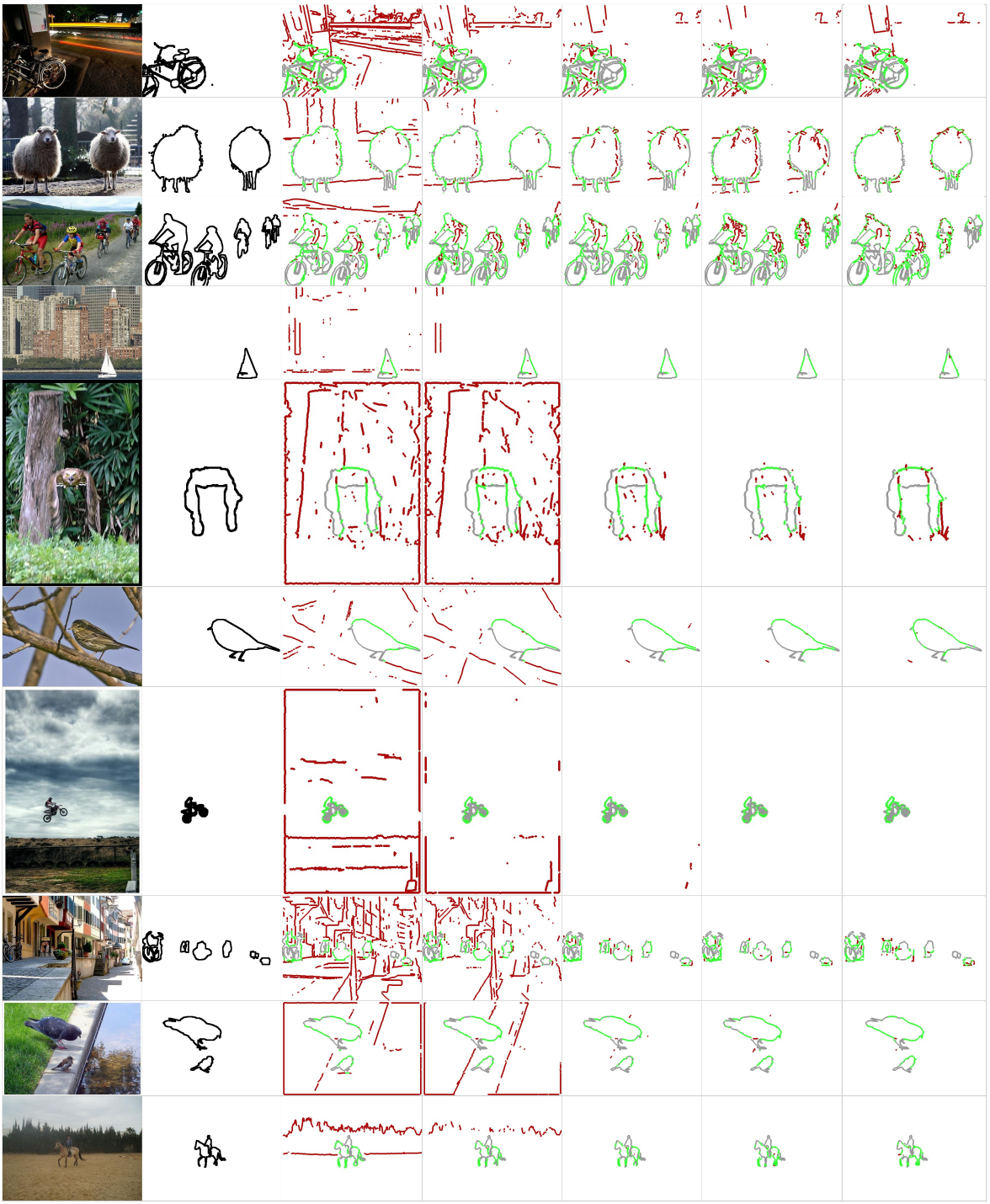


Figure 3.9: Qualitative results on VOC12. $(\cdot)$ denotes the data used for training. Red/green indicate false/true positive pixels, grey is missing recall. All methods are shown at 50% recall. Det.+SE (weak) refers to the model Det.+SE ($\text{SeSe}_+ \cap \text{BBs}$) Det.+HED (weak) refers to Det.+HED (cons. S&G $\cap$ BBs). Object-specific boundaries differ from generic boundaries (such as the ones detected by SE(BSDS)). By using an object detector we can suppress non-object boundaries and focus boundary detection on the classes of interest. The proposed weakly supervised techniques allow to achieve high quality boundary estimates that are similar to the ones obtained by fully supervised methods.

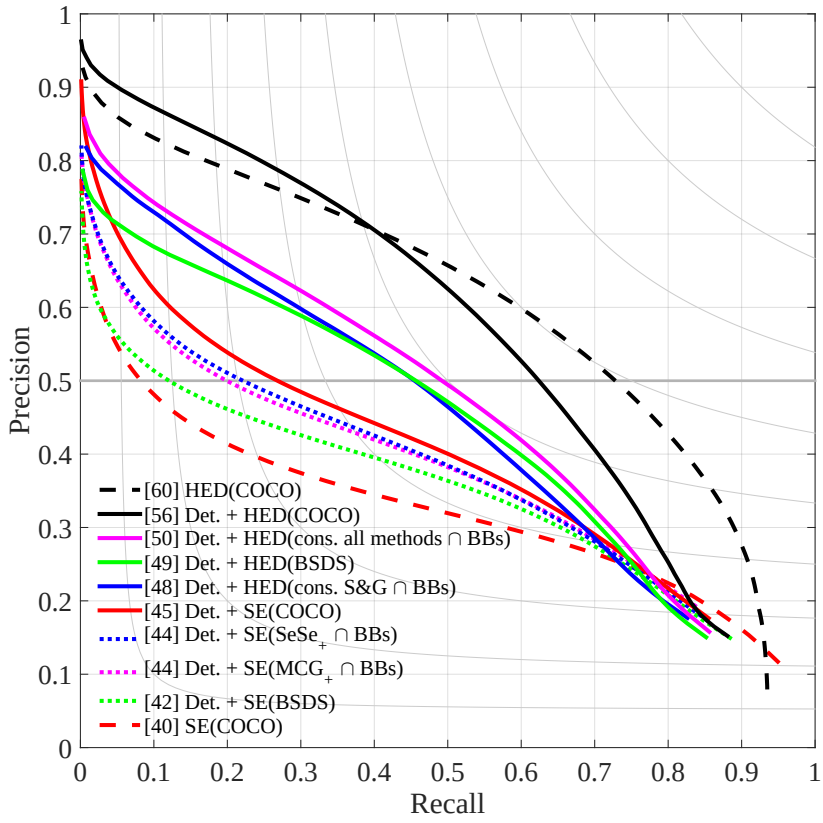


Figure 3.10: COCO results. $(\cdot)$ denotes the data used for training. Continuous/dashed line indicates models using/not using a detector at test time. Legend indicates AP numbers. For weakly supervised cases the results are shown with the models trained on VOC, without re-training on COCO.

3.7 COCO BOUNDARY DETECTION RESULTS

Additionally we show the generalization of the proposed weakly supervised variants for object boundary detection on the COCO dataset. We use the same evaluation protocol as for VOC. For weakly supervised cases the results are shown with the models trained on VOC, without re-training on COCO.

The results are summarized in Table 3.4 and in Figure 3.10. On the COCO benchmark for both SE and HED the models trained on the proposed weak annotations perform as well as the fully supervised SE models. Similar to the VOC benchmark the HED model trained on ground truth shows superior performance.

3.8 SBD BOUNDARY DETECTION RESULTS

In this section we analyse the performance of the proposed weakly supervised boundary variants trained with SE and HED on the SBD dataset (Hariharan *et al.*,

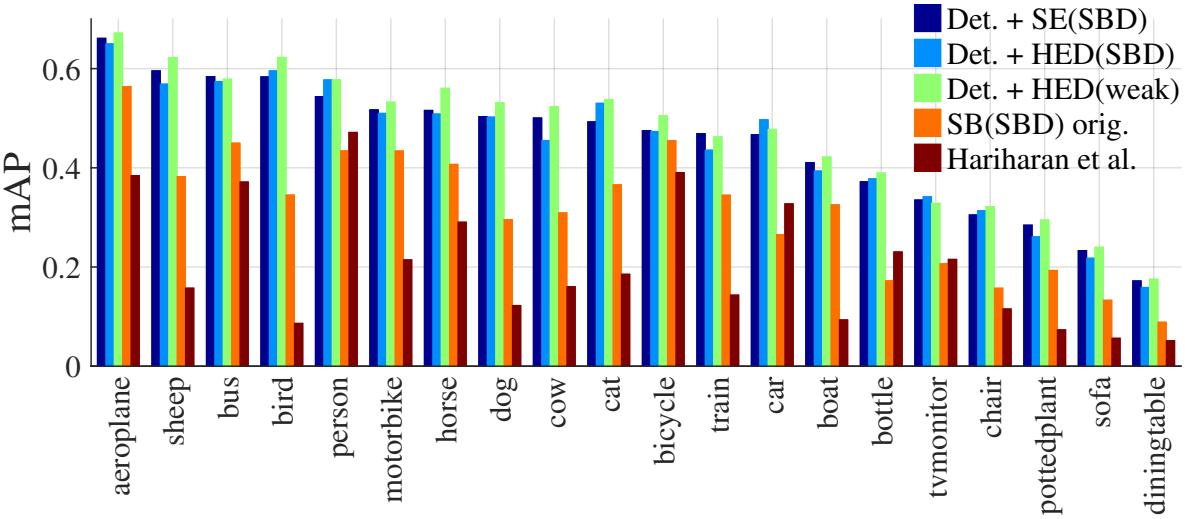


Figure 3.11: SBD results per class. $(\cdot)$ denotes the data used for training. Det.+ HED (weak) refers to the model Det.+HED (cons. S&G $\cap$ BBs).

2011). In contrast to the VOC benchmark we move from object boundaries to class specific object boundaries. We are interested in external boundaries of all annotated objects of the specific semantic class and all internal boundaries are ignored during evaluation following the benchmark (Hariharan *et al.*, 2011). The results are presented in Figure 3.11 and in Table 3.5.

Fully supervised. Applying SE model plus object detection at test time outperforms the class specific situational boundary detector (for both Uijlings and Ferrari (2015) and our re-implementation) as well as the Inverse Detectors (Hariharan *et al.*, 2011). The model trained with SE on ground truth performs as well as the HED detector. Both of the models are good at detecting external object boundaries; however SE, being a more local, triggers more on internal boundaries than HED. In the VOC evaluation detecting internal object boundaries is penalized, while in SBD these are ignored. This explains the small gap in the performance between SE and HED on this benchmark.

Weakly supervised. The models trained with the proposed weakly-supervised boundary variants perform on par with the fully supervised detectors, while only using bounding boxes or generic boundary annotations. We show in Table 3.5 the top result with the Det. + HED(cons. S&G$\cap$BBs) model, achieving the state-of-the-art performance on the SBD benchmark. As Figure 3.11 shows our weakly supervised approach considerably outperforms Uijlings and Ferrari (2015) and Hariharan *et al.* (2011) on all 20 classes.

	Family	Method	mF	mAP
Other	GT	Hariharan et al. (Hariharan *et al.*, 2011)	28	21
SE	GT	SB(SBD) orig. (Uijlings and Ferrari, 2015)	39	32
		SB(SBD)	43	37
		Det.+SE (SBD)	51	45
	Other GT	Det.+SE (BSDS)	51	44
		Det.+MCG (BSDS)	50	42
	Weakly supervised	SB(SeSe ∩ BBs)	40	34
		SB (MCG ∩ BBs)	42	35
		Det.+SE (SeSe ∩ BBs)	48	42
		Det.+SE (MCG ∩ BBs)	**51**	**45**
HED	GT	HED (SBD)	44	41
		Det.+HED (SBD)	49	45
	Other GT	HED(BSDS)	38	32
		Det.+HED (BSDS)	49	44
	Weakly supervised	HED(cons. MCG ∩ BBs)	41	37
		HED (cons. S&G ∩ BBs)	44	39
		Det.+HED (cons. MCG ∩ BBs)	48	44
		Det.+HED (cons. S&G ∩ BBs)	**52**	**47**

Table 3.5: SBD results. Results are mean F(ODS)/AP across all 20 categories. $(\cdot)$ denotes the data used for training. See also Figure 3.11. Underline indicates ground truth baselines, and bold are our best weakly supervised results.

3.9 CONCLUSION

The presented experiments show that high quality object boundaries can be achieved using only detection bounding box annotations. With these alone, our proposed weak-supervision techniques already improve over previously reported fully supervised results for object-specific boundaries. When using generic boundary or ground truth annotations, we achieve the top performance on the object boundary detection task, outperforming previously reported results by a large margin.

In Chapter 4 we extend the proposed approach to other closely related tasks, such as semantic labelling and instance segmentation.

4 SIMPLE DOES IT: WEAKLY SUPERVISED INSTANCE AND SEMANTIC SEGMENTATION

In Chapter 3 we addressed the problem of weakly supervised object boundary detection. Semantic labelling and instance segmentation are another two tasks that require particularly costly pixel-level annotations and are in need of relaxing this constraint.

Similarly to Chapter 3, we employ weak supervision in the form of bounding box detection annotations and propose an approach that does not require modification of the segmentation training procedure. We show that when carefully designing the input labels from given bounding boxes, even a single round of training is enough to improve over previously reported weakly supervised results. Overall, our weak supervision approach reaches $\sim$95% of the quality of the fully supervised model, both for semantic labelling and instance segmentation.

4.1 INTRODUCTION

Convolutional networks (convnets) have become the de facto technique for pattern recognition problems in computer vision. One of their main strengths is the ability to profit from extensive amounts of training data to reach top quality. However, one of their main weaknesses is that they need a large number of training samples for high quality results. This is usually mitigated by using pre-trained models (e.g. with $\sim 10^6$ training samples for ImageNet classification (Russakovsky *et al.*, 2015)), but still thousands of samples are needed to shift from the pre-training domain to the application domain. Applications such as semantic labelling (associating each image pixel to a given class) or instance segmentation (grouping all pixels belonging to the same object instance) are expensive to annotate, and thus significant cost is involved in creating large enough training sets.

Compared to object bounding box annotations, pixel-wise mask annotations are far more expensive, requiring $\sim 15\times$ more time (Lin *et al.*, 2014). Cheaper and easier to define, box annotations are more pervasive than pixel-wise annotations. In principle, a large number of box annotations (and images representing the background class) should convey enough information to understand which part of the box content is foreground and which is background. In this chapter we explore how much one can close the gap between training a convnet using full supervision for semantic labelling (or instance segmentation) versus using only bounding box annotations.

Our experiments focus on the 20 Pascal classes (Everingham *et al.*, 2015) and show that using only bounding box annotations over the same training set we

Training sample, with box annotations | Test image, fully supervised result | Test image, weakly supervised result

Figure 4.1: We propose a technique to train semantic labelling from bounding boxes, and reach 95% of the quality obtained when training from pixel-wise annotations.

can reach $\sim$95% of the accuracy achievable with full supervision. We show top results for (bounding box) weakly supervised semantic labelling and, to the best of our knowledge, for the first time report results for weakly supervised instance segmentation.

We view the problem of weak supervision as an issue of input label noise. We explore recursive training as a de-noising strategy, where convnet predictions of the previous training round are used as supervision for the next round. We also show that, when properly used, "classic computer vision" techniques for box-guided instance segmentation are a source of surprisingly effective supervision for convnet training.

In summary, our main contributions are:

- We explore recursive training of convnets for weakly supervised semantic labelling, discuss how to reach good quality results, and what are the limitations of the approach (Section 4.2.1).
- We show that state-of-the-art quality can be reached when properly employing GrabCut-like algorithms to generate training labels from given bounding boxes, instead of modifying the segmentation convnet training procedure (Section 4.2.2).
- We report the best known results when training using bounding boxes only, both using Pascal VOC12 and VOC12+COCO training data, reaching comparable quality with the fully supervised regime (Section 4.3.2).
- We are the first to show that similar results can be achieved for the weakly supervised instance segmentation task (Section 4.5).

4.2 FROM BOXES TO SEMANTIC LABELS

The goal of this work is to provide high quality semantic labelling starting from object bounding box annotations. We design our approach aiming to exploit the

available information at its best. There are two sources of information: the annotated boxes and priors about the objects. We integrate these in the following cues:

C1 Background. Since the bounding boxes are expected to be exhaustive, any pixel not covered by a box is labelled as background.

C2 Object extent. The box annotations bound the extent of each instance. Assuming a prior on the objects shapes (e.g. oval-shaped objects are more likely than thin bar or full rectangular objects), the box also gives information on the expected object area. We employ this size information during training.

C3 Objectness. In addition to extent and area, there are other object priors at hand. Two priors typically used are spatial continuity and having a contrasting boundary with the background. In general we can harness priors about object shape by using segment proposal techniques (Pont-Tuset and Gool, 2015), which are designed to enumerate and rank plausible object shapes in an area of the image.

4.2.1 Box baselines

We first describe a naive baseline that serves as starting point for our exploration. Given an annotated bounding box and its class label, we label all pixels inside the box with such given class. If two boxes overlap, we assume the smaller one is in front. Any pixel not covered by boxes is labelled as background.

Figure 4.2 left side and Figure 4.3c show such example annotations. We use these labels to train a segmentation network with the standard training procedure. We employ the DeepLabv1 approach from Chen *et al.* (2015) (details in Section 4.3.1).

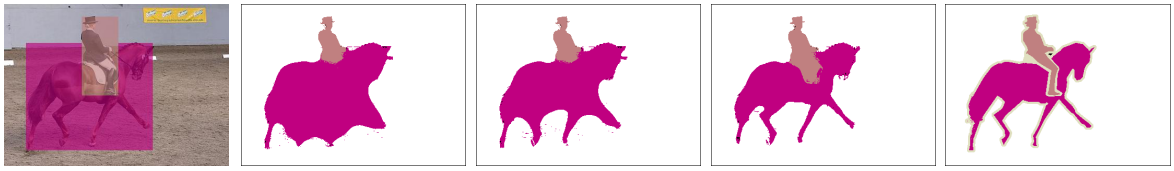
Recursive training. We observe that when applying the resulting model over the training set, the network outputs capture the object shape significantly better than just boxes (see Figure 4.2). This inspires us to follow a recursive training procedure, where these new labels are fed in as ground truth for a second training round. We name this recursive training approach `Naive`.

The recursive training is enhanced by de-noising the convnet outputs using extra information from the annotated boxes and object priors. Between each round we improve the labels with three post-processing stages:

1. Any pixel outside the box annotations is reset to background label (cue C1).

2. If the area of a segment is too small compared to its corresponding bounding box (e.g. IoU$<$ 50%), the box area is reset to its initial label (fed in the first round). This enforces a minimal area (cue C2).

3. As it is common practice among semantic labelling methods, we filter the output of the network to better respect the image boundaries. (We use DenseCRF (Krähenbühl and Koltun, 2011) with the DeepLabv1 parameters (Chen



Example input rectangles | Output after 1 training round | After 5 rounds | After 10 rounds | Ground truth

Figure 4.2: Example results of using only rectangle segments and recursive training (using convnet predictions as supervision for the next round), see Section 4.2.1.

et al., 2015)). In our weakly supervised scenario, boundary-aware filtering is particularly useful to improve objects delineation (cue C3).

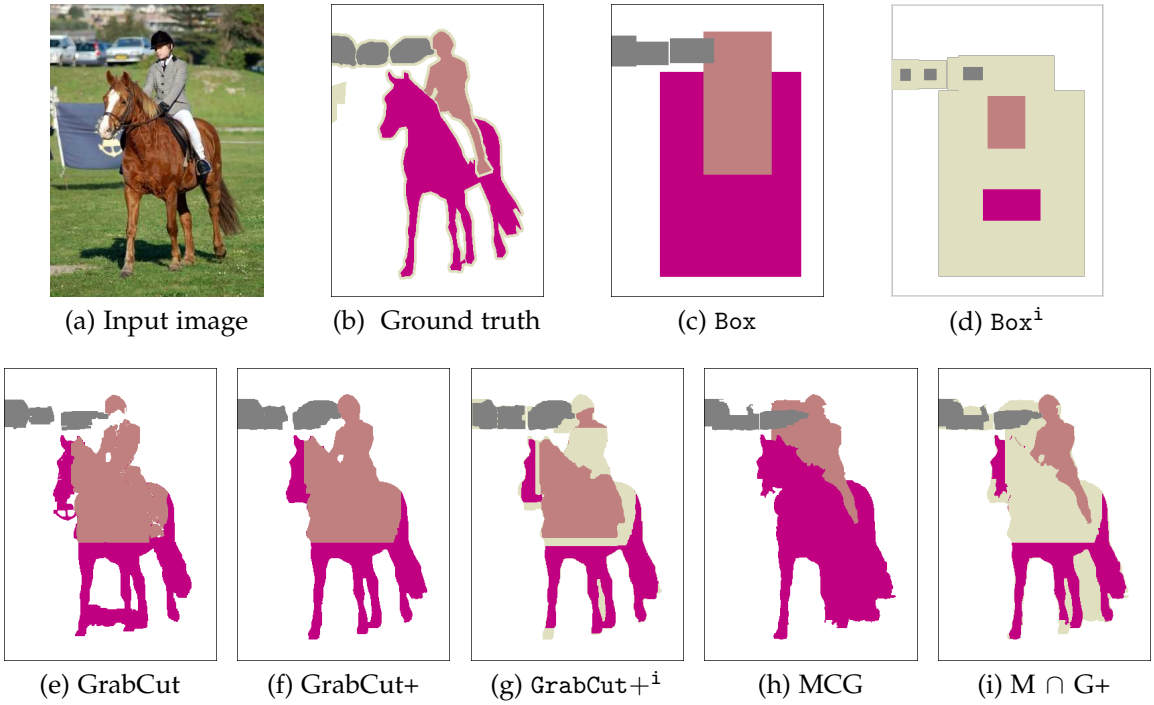
The recursion and these three post-processing stages are crucial to reach good performance. We name this recursive training approach `Box`, and show an example result in Figure 4.2.

Ignore regions. We also consider a second variant $\texttt{Box}^{\texttt{i}}$ that, instead of using filled rectangles as initial labels, we fill in the 20% inner region, and leave the remaining inner area of the bounding box as ignore regions. See Figure 4.3d. Following cues C2 and C3 (shape and spatial continuity priors), the 20% inner box region should have higher chances of overlapping with the corresponding object, reducing the noise in the generated input labels. The intuition is that the convnet training might benefit from trading-off lower recall (more ignore pixels) for higher precision (more pixels are correctly labelled). Starting from this initial input, we use the same recursive training procedure as for `Box`. Despite the simplicity of the approach, as we will see in the experimental section 4.3, `Box` / $\texttt{Box}^{\texttt{i}}$ are already competitive with the current state of the art.

However, using rectangular shapes as training labels is clearly suboptimal. Therefore, in the next section, we propose an approach that obtains better results while avoiding multiple recursive training rounds.

4.2.2 Box-driven segments

The box baselines are purposely simple. A next step in complexity consists in utilising the box annotations to generate an initial guess of the object segments. We think of this as "old school meets new school": we use the noisy outputs of classic computer vision methods, box-driven figure-ground segmentation (Rother *et al.*, 2004) and object proposal (Pont-Tuset and Gool, 2015) techniques, to feed the training of a convnet. Although the output object segments are noisy, they are more precise than simple rectangles, and thus should provide improved results. A single training round will be enough to reach good quality.



(a) Input image (b) Ground truth (c) Box (d) Box^{i}

(e) GrabCut (f) GrabCut+ (g) GrabCut+^{i} (h) MCG (i) M ∩ G+

Figure 4.3: Example of the different segmentations obtained starting from a bounding box annotation. Grey/pink/magenta indicate different object classes, white is background, and ignore regions are beige. M ∩ G+ denotes MCG ∩ GrabCut+.

4.2.2.1 *GrabCut baselines*

GrabCut (Rother *et al.*, 2004) is an established technique to estimate an object segment from its bounding box. We propose to use a modified version of GrabCut, which we call GrabCut+, where HED boundaries (Xie and Tu, 2015) are used as pairwise term instead of the typical RGB colour difference. (The HED boundary detector is trained on the generic boundaries of BSDS500 (Arbeláez *et al.*, 2011)). We considered other GrabCut variants, such as Cheng *et al.* (2015a); Tang *et al.* (2015). However, the proposed GrabCut+ gives higher quality segments (75.2 mIoU compared to 73.5 mIoU (Tang *et al.*, 2015) and 52.5 mIoU (Cheng *et al.*, 2015a) on the Pascal VOC12 validation set).

Similar to Box^{i}, we also consider a GrabCut+^{i} variant, which trades off recall for higher precision. For each annotated box we generate multiple (∼150) perturbed GrabCut+ outputs. If 70% of the segments mark the pixel as foreground, the pixel is set to the box object class. If less than 20% of the segments mark the pixels as foreground, the pixel is set as background, otherwise it is marked as ignore. The perturbed outputs are generated by jittering the box coordinates (±5%) as well as the size of the outer background region considered by GrabCut (from 10% to 60%). An example result of GrabCut+^{i} can be seen in Figure 4.3g.

Figure 4.4: More examples of segmentation annotations obtained starting from a bounding box. White is background and ignore regions are beige. M ∩ G+ denotes MCG ∩ Grabcut+.

4.2.2.2 *Adding objectness*

With our final approach we attempt to better incorporate the object shape priors by using segment proposals (Pont-Tuset and Gool, 2015). Segment proposals techniques are designed to generate a soup of likely object segmentations, incorporating as many "objectness" priors as useful (cue C3).

We use the state-of-the-art proposals from MCG (Pont-Tuset *et al.*, 2016). As the final stage the MCG algorithm includes a ranking based on a decision forest trained over the Pascal VOC 2012 dataset. We do *not* use this last ranking stage, but instead use all the (unranked) generated segments. Given a box annotation, we pick the highest overlapping proposal as a corresponding segment.

Building upon the insights from the baselines in Section 4.2.1 and 4.2.2, we use the MCG segment proposals to supplement GrabCut+. Inside the annotated boxes, we mark as foreground pixels where both MCG and GrabCut+ agree; the remaining ones are marked as ignore. We denote this approach as $\texttt{MCG} \cap \texttt{GrabCut+}$ or $\texttt{M} \cap \texttt{G+}$ for short. Because MCG and GrabCut+ provide complementary information, we can think of $\texttt{M} \cap \texttt{G+}$ as an improved version of $\texttt{GrabCut+}^{\texttt{i}}$ providing a different trade-off between precision and recall on the generated labels (see Figure 4.3i). More examples of generated segmentation annotations can be seen in Figure 4.4.

The BoxSup method (Dai *et al.*, 2015a) also uses MCG object proposals during training; however, there are important differences. They modify the training procedure so as to denoise intermediate outputs by randomly selecting high overlap proposals. In comparison, our approach keeps the training procedure unmodified and simply generates input labels. Our approach also uses ignore regions, while BoxSup does not explore this dimension. Finally, BoxSup uses a longer training than our approach.

Section 4.3 shows results for the semantic labelling task, compares different methods and different supervision regimes. In Section 4.4 we show that the proposed approach is also suitable for the instance segmentation task.

4.3 SEMANTIC LABELLING RESULTS

Our approach is equally suitable (and effective) for weakly supervised instance segmentation as well as for semantic labelling. However, only the latter has directly comparable related work. We thus focus our experimental comparison efforts on the semantic labelling task. Results for instance segmentation are presented in Section 4.5.

Section 4.3.1 discusses the experimental setup, evaluation, and implementation details for semantic labelling. Section 4.3.2 presents our main results, contrasting the methods from Section 4.2 with the current state of the art. Section 4.3.3 further expands these results with a more detailed analysis, and presents results when using more supervision (semi-supervised case).

4.3.1 Experimental setup

Datasets. We evaluate the proposed methods on the Pascal VOC12 segmentation benchmark (Everingham *et al.*, 2015). The dataset consists of 20 foreground object classes and one background class. The segmentation part of the VOC12 dataset contains 1 464 training, 1 449 validation, and 1 456 test images. Following previous work (Chen *et al.*, 2015; Dai *et al.*, 2015a), we extend the training set with the annotations provided by Hariharan *et al.* (2011), resulting in an augmented set of 10 582 training images.

In some of our experiments, we use additional training images from the COCO (Lin *et al.*, 2014) dataset. We only consider images that contain any of the 20 Pascal classes and (following Zheng *et al.* (2015)) only objects with a bounding box area larger than 200 pixels. After this filtering, 99 310 images remain (from training and validation sets), which are added to our training set. When using COCO data, we first pre-train on COCO and then fine-tune over the Pascal VOC12 training set. All of the COCO and Pascal training images come with semantic labelling annotations (for fully supervised case) and bounding box annotations (for weakly supervised case).

Evaluation. We use the "comp6" evaluation protocol. The performance is measured in terms of pixel intersection-over-union averaged across 21 classes (mIoU). Most of our results are shown on the validation set, which we use to guide our design choices. Final results are reported on the test set (via the evaluation server) and compared with other state-of-the-art methods.

Implementation details. For all our experiments we use the DeepLab-LargeFOV network, using the same train and test parameters as Chen *et al.* (2015). The model is initialized from a VGG16 network pre-trained on ImageNet (Simonyan and Zisserman, 2015). We use a mini-batch of 30 images for SGD and initial learning rate of 0.001, which is divided by 10 after a 2k/20k iterations (for Pascal/COCO). At test time, we apply DenseCRF (Krähenbühl and Koltun, 2011). Our network and post-processing are comparable to the ones used in Dai *et al.* (2015a); Papandreou *et al.* (2015).

Note that multiple strategies have been considered to boost test time results, such as multi-resolution or model ensembles (Chen *et al.*, 2015; Kokkinos, 2016). Here we keep the approach simple and fixed. In all our experiments we use a fixed training and test time procedure. Across experiments we only change the input training data that the networks gets to see.

4.3.2 Main results

Box results. Figure 4.5 presents the results for the recursive training of the box baselines from Section 4.2.1. We see that the `Naive` scheme, a recursive training from rectangles disregarding post-processing stages, leads to poor quality. However,

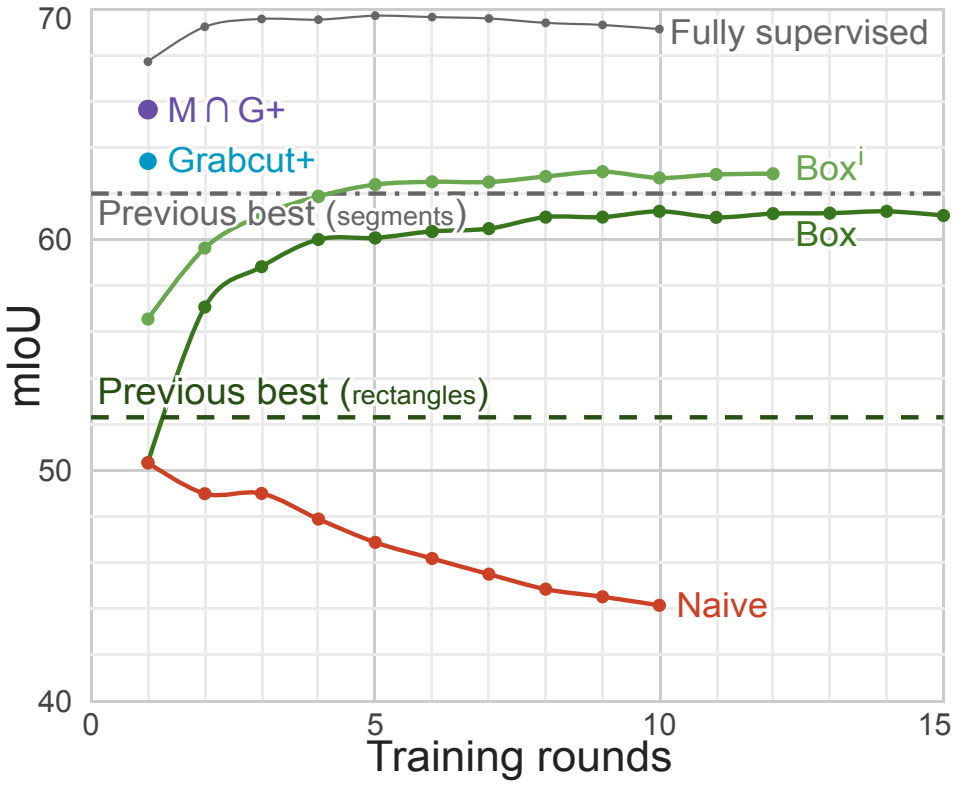


Figure 4.5: Segmentation quality versus training round for different approaches, see also Tables 4.1 and 4.2. Pascal VOC12 validation set results. "Previous best (rectangles/segments)" corresponds to WSSL_R/BoxSup_MCG in Table 4.2.

Method		val. mIoU
-	Fast-RCNN	44.3
	GT Boxes	62.2
Weakly supervised	Box	61.2
	Box^i	62.7
	MCG	62.6
	GrabCut+	63.4
	GrabCut+^{i}	64.3
	M ∩ G+	**65.7**
Fully supervised	DeepLab$_{ours}$ (Chen *et al.*, 2015)	69.1

Table 4.1: Weakly supervised semantic labelling results for our baselines. Trained using Pascal VOC12 bounding boxes alone, validation set results. DeepLab$_{ours}$ indicates our fully supervised result.

Super-vision	#GT images	#Weak images	Method	val. set mIoU	test set mIoU	test set FS%
VOC12 (V)						
Weak	-	V 10k	Bearman et al. (Bearman *et al.*, 2015)	45.1	-	-
			BoxSup_{R} (Dai *et al.*, 2015a)	52.3	-	-
			WSSL_{R}(Papandreou *et al.*, 2015)	52.5	54.2	76.9
			WSSL_{S}(Papandreou *et al.*, 2015)	60.6	62.2	88.2
			$\text{BoxSup}_{\text{MCG}}$(Dai *et al.*, 2015a)	62.0	64.6	91.6
			$\texttt{Box}^{\texttt{i}}$	62.7	63.5	90.0
			$\texttt{M} \cap \texttt{G}+$	**65.7**	**67.5**	**95.7**
Semi	V 1.4k	V 9k	WSSL_{R}(Papandreou *et al.*, 2015)	62.1	-	-
			$\text{BoxSup}_{\text{MCG}}$(Dai *et al.*, 2015a)	63.5	66.2	93.9
			WSSL_{S}(Papandreou *et al.*, 2015)	65.1	66.6	94.5
			$\texttt{M} \cap \texttt{G}+$	**65.8**	**66.9**	**94.9**
Full	V 10k	-	BoxSup (Dai *et al.*, 2015a)	63.8	-	-
			WSSL (Papandreou *et al.*, 2015)	67.6	70.3	99.7
			DeepLab_{ours} (Chen *et al.*, 2015)	<u>69.1</u>	<u>70.5</u>	100
VOC12 + COCO (V+C)						
Weak	-	V+C 110k	$\texttt{Box}^{\texttt{i}}$	65.3	66.7	91.1
			$\texttt{M} \cap \texttt{G}+$	**68.9**	**69.9**	**95.5**
Semi	V 10k	C 123k C 100k	$\text{BoxSup}_{\text{MCG}}$(Dai *et al.*, 2015a)	68.2	71.0	97.0
			$\texttt{M} \cap \texttt{G}+$	**71.6**	**72.8**	**99.5**
Full	V+C 133k V+C 110k	-	BoxSup (Dai *et al.*, 2015a)	68.1	-	-
			WSSL (Papandreou *et al.*, 2015)	71.7	73	99.7
			DeepLab_{ours} (Chen *et al.*, 2015)	<u>72.3</u>	<u>73.2</u>	100

Table 4.2: Semantic labelling results for validation and test set; under different training regimes with VOC12 (V) and COCO data (C). Underline indicates full supervision baselines, and bold are our best weakly- and semi-supervised results. FS%: performance relative to the best fully supervised model (DeepLab_{ours}). Discussion in Sections 4.3.2 and 4.3.3.

Supervision	Method	mIoU	FS%
VOC12			
Weak	$\texttt{M} \cap \texttt{G}+$	69.4	93.2
Full	DeepLabv2-ResNet101 (Chen *et al.*, 2016b)	<u>74.5</u>	100
VOC12 + COCO			
Weak	$\texttt{M} \cap \texttt{G}+$	74.2	95.5
Full	DeepLabv2-ResNet101 (Chen *et al.*, 2016b)	<u>77.7</u>	100

Table 4.3: DeepLabv2-ResNet101 network semantic labelling results on VOC12 validation set, using VOC12 or VOC12+COCO training data. FS%: performance relative to the full supervision. Discussion in Section 4.3.3.

Figure 4.6: Qualitative results on VOC12. Visually, the results from our weakly supervised method M ∩ G+ are hardly distinguishable from the fully supervised ones.

by using the suggested three post-processing stages, the Box baseline obtains a significant gain, getting tantalisingly close to the best reported results on the task (Dai *et al.*, 2015a). Adding ignore regions inside the rectangles (Box $\rightarrow$ Box$^{\text{i}}$) provides a clear gain and leads by itself to state-of-the-art results.

Figure 4.5 also shows the result of using longer training for fully supervised case. When using ground truth semantic segmentation annotations, one training round is enough to achieve good performance; longer training brings marginal improvement. As discussed in Section 4.2.1, reaching good quality for Box/Box$^{\text{i}}$ requires multiple training rounds instead, and performance becomes stable from round 5 onwards. Instead, GrabCut+/M$\cap$G+ do not benefit from additional training rounds.

Box-driven segment results. Table 4.1 evaluates results on the Pascal VOC12 validation set. It indicates the Box/Box$^{\text{i}}$ results after 10 rounds, and MCG/GrabCut+/GrabCut+$^{\text{i}}$/M$\cap$G+ results after one round. "Fast-RCNN" is the result using detections (Girshick, 2015) to generate semantic labels (lower-bound), "GT Boxes" considers the box annotations as labels, and DeepLab$_{ours}$ indicates our fully supervised segmentation network result obtained with a training length equivalent to three training rounds (upper-bound for our results). We see in the results that using ignore regions systematically helps (trading-off recall for precision), and that M$\cap$G+ provides better results than MCG and GrabCut+ alone.

Table 4.2 indicates the box-driven segment results after 1 training round and shows comparison with other state-of-the-art methods, trained from boxes only using either Pascal VOC12, or VOC12+COCO data. BoxSup$_{\text{R}}$ and WSSL$_{\text{R}}$ both feed the network with rectangle segments (comparable to Box$^{\text{i}}$), while WSSL$_{\text{S}}$ and BoxSup$_{\text{MCG}}$ exploit arbitrary shaped segments (comparable to M$\cap$G+). Although our network and post-processing is comparable to the ones in Dai *et al.* (2015a); Papandreou *et al.* (2015), there are differences in the exact training procedure and parameters.

Overall, our results indicate that - without modifying the training procedure - M$\cap$G+ is able to improve over previously reported results and reach 95% of the fully-supervised training quality. By training with COCO data (Lin *et al.*, 2014) before fine-tuning for Pascal VOC12, we see that with enough additional bounding boxes we can match the full supervision from Pascal VOC 12 (68.9 versus 69.1). This shows that the labelling effort could be significantly reduced by replacing segmentation masks with bounding box annotations.

4.3.3 Additional results

Semi-supervised case. Table 4.2 compares results in the semi-supervised modes considered by Dai *et al.* (2015a); Papandreou *et al.* (2015), where some of the images have full supervision, and some have only bounding box supervision. Training with 10% of Pascal VOC12 semantic labelling annotations does not bring much gain to the performance (65.7 versus 65.8), this hints at the high quality of the generated M$\cap$G+ input data.

By using ground-truth annotations on Pascal plus bounding box annotations on COCO, we observe 2.5 points gain (69.1 $\rightarrow$ 71.6 , see Table 4.2). This suggests that the overall performance could be further improved by using extra training data with bounding box annotations.

Boundaries supervision. Our results from MCG, GrabCut+, and M $\cap$ G+ all indirectly include information from the BSDS500 dataset (Arbeláez *et al.*, 2011) via the HED boundary detector (Xie and Tu, 2015). These results are fully comparable to BoxSup-MCG (Dai *et al.*, 2015a), to which we see a clear improvement. Nonetheless one would like to know how much using dense boundary annotations from BSDS500 contributes to the results. We use the weakly supervised boundary detection technique from Khoreva *et al.* (2016b) to learn boundaries directly from the Pascal VOC12 box annotations. Training M $\cap$ G+ using weakly supervised HED boundaries results in 1 point loss compared to using the BSDS500 (64.8 versus 65.7 mIoU on Pascal VOC12 validation set). We see then that although the additional supervision does bring some help, it has a minor effect and our results are still rank at the top even when we use only Pascal VOC12 + ImageNet pre-training.

Different convnet results. For comparison purposes with Dai *et al.* (2015a); Papandreou *et al.* (2015) we used DeepLabv1 with a VGG-16 network in our experiments. To show that our approach also generalizes across different convnets, we also trained DeepLabv2 with a ResNet101 network (Chen *et al.*, 2016b). Table 4.3 presents the results.

Similar to the case with VGG-16, our weakly supervised approach M $\cap$ G+ reaches 93%/95% of the fully supervised case when training with VOC12/VOC12+COCO, and the weakly supervised results with COCO data reach similar quality to full supervision with VOC12 only.

4.4 FROM BOXES TO INSTANCE SEGMENTATION

Complementing the experiments of the previous sections, we also explore a second task: weakly supervised instance segmentation. To the best of our knowledge, these are the first reported experiments on this task.

As object detection moves forward, there is a need to provide richer output than a simple bounding box around objects. Recently (Hariharan *et al.*, 2015; Pinheiro *et al.*, 2015, 2016) explored training convnets to output a foreground versus background segmentation of an instance inside a given bounding box. Such networks are trained using pixel-wise annotations that distinguish between instances. These annotations are more detailed and expensive than semantic labelling, and thus there is interest in weakly supervised training.

The segments used for training, as discussed in Section 4.2.2, are generated starting from individual object bounding boxes. Each segment represents a different object instance and thus can be used directly to train an instance segmentation

Figure 4.7: Example result from our weakly supervised DeepMask (VOC12+COCO) model.

convnet. For each annotated bounding box, we generate a foreground versus background segmentation using the GrabCut+ method (Section 4.2.2), and train a convnet to regress from the image and bounding box information to the instance segment.

4.5 INSTANCE SEGMENTATION RESULTS

Experimental setup. We choose a purposely simple instance segmentation pipeline, based on the "hyper-columns system 2" architecture (Hariharan *et al.*, 2015). We use Fast-RCNN (Girshick, 2015) detections (post-NMS) with their class score, and for each detection estimate an associated foreground segment. We estimate the foreground using either some baseline method (e.g. GrabCut) or using convnets trained for the task (Pinheiro *et al.*, 2015; Chen *et al.*, 2016b).

For our experiments we use a re-implementation of the DeepMask (Pinheiro *et al.*, 2015) architecture, and additionally we re-purpose a DeepLabv2 VGG-16 network (Chen *et al.*, 2016b) for the instance segmentation task, which we name DeepLab$_{\text{BOX}}$. Inspired by Xu *et al.* (2016); Carreira *et al.* (2016), we modify DeepLab to accept four input channels: the input image RGB channels, plus a binary map with a bounding box of the object instance to segment. We train the network DeepLab$_{\text{BOX}}$ to output the segmentation mask of the object corresponding to the input bounding box. The additional input channel guides the network so as to segment only the instance of interest instead of all objects in the scene. The input box rectangle can also be seen as an initial guess of the desired output. We train using ground truth bounding boxes, and at test time Fast-RCNN detection boxes are used.

We train DeepMask and DeepLab$_{\text{BOX}}$ using GrabCut+ results either over Pascal VOC12 or VOC12+COCO data (1 training round, no recursion like in Section 4.2.1), and test on the VOC12 validation set, the same set of images used in Section 4.3. The augmented annotation from Hariharan *et al.* (2011) provides per-instance segments for VOC12. We do not use CRF post-processing for neither of the networks.

Following instance segmentation literature (Hariharan *et al.*, 2014, 2015) we report in Table 4.4 mAP^r at IoU threshold 0.5 and 0.75. mAP^r is similar to the traditional

DeepMask

$\text{DeepLab}_{\text{BOX}}$

Figure 4.8: Example results from the DeepMask and $\text{DeepLab}_{\text{BOX}}$ models trained with Pascal VOC12 and COCO using box supervision. White boxes illustrate Fast-RCNN detection proposals used to output the segments which have the best overlap with the ground truth segmentation mask.

Supervision	Method	$mAP^r_{0.5}$	$mAP^r_{0.75}$	ABO
-	Rectangle	21.6	1.8	38.5
	Ellipse	29.5	3.9	41.7
	MCG	28.3	5.9	44.7
	GrabCut	38.5	13.9	45.8
	GrabCut+	41.1	17.8	46.4
VOC12				
Weak	DeepMask	39.4	8.1	45.8
	$DeepLab_{BOX}$	44.8	16.3	**49.1**
Full	DeepMask	41.7	9.7	47.1
	$DeepLab_{BOX}$	47.5	20.2	<u>51.1</u>
VOC12 + COCO				
Weak	DeepMask	42.9	11.5	48.8
	$DeepLab_{BOX}$	46.4	18.5	**51.4**
Full	DeepMask	44.7	13.1	49.7
	$DeepLab_{BOX}$	49.4	23.7	<u>53.1</u>

Table 4.4: Instance segmentation results on VOC12 validation set. Underline indicates the full supervision baseline, and bold are our best weak supervision results. Weakly supervised DeepMask and $DeepLab_{BOX}$ reach comparable results to full supervision. See Section 4.5 for details.

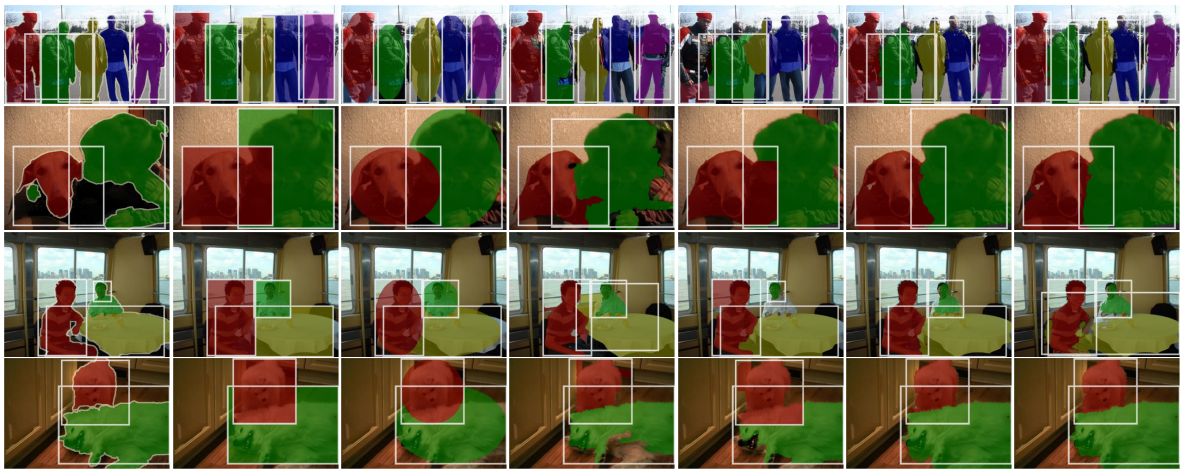


Figure 4.9: Qualitative results of instance segmentation on VOC12. Example result from the DeepMask model are trained with Pascal VOC12 and COCO supervision. White boxes illustrate Fast-RCNN detection proposals used to output the segments which have the best overlap with the ground truth segmentation mask.

VOC12 evaluation, but using IoU between segments instead of between boxes. Since we have a fixed set of windows, we can also report the average best overlap (ABO) (Pont-Tuset and Gool, 2015) metric to give a different perspective on the results.

Baselines. We consider five training-free baselines: simply filling in the detection rectangles (boxes) with foreground labels, fitting an ellipse inside the box, using the MCG proposal with best bounding box IoU, and using GrabCut and `GrabCut+` (see Section 4.2.2 and Figure 4.9), initialized from the detection box.

Analysis. The results table 4.4 follows the same trend as the semantic labelling results in Section 4.3. `GrabCut+` provides the best results among the baselines considered and shows comparable performance to DeepMask, while our proposed $\text{DeepLab}_{\text{BOX}}$ outperforms both techniques. We see that our weakly supervised approach reaches $\sim$95% of the quality of fully-supervised case (both on $\text{mAP}^r_{0.5}$ and ABO metrics) using two different convnets, DeepMask and $\text{DeepLab}_{\text{BOX}}$, both when training with VOC12 or VOC12+COCO. Examples of the instance segmentation results from weakly supervised DeepMask and $\text{DeepLab}_{\text{BOX}}$ are shown in Figure 4.8.

4.6 CONCLUSION

The series of experiments presented in this chapter provides new insights on how to train pixel-labelling convnets from bounding box annotations only. We showed that when carefully employing the available cues, recursive training using only rectangles as input can be surprisingly effective ($\texttt{Box}^{\texttt{i}}$). Even more, when using box-driven segmentation techniques and doing a good balance between accuracy and recall in the noisy training segments, we can reach state-of-the-art performance without modifying the segmentation network training procedure ($\texttt{M} \cap \texttt{G+}$). Our results improve over previously reported ones on the semantic labelling task and reach $\sim$95% of the quality of the same network trained on the ground truth segmentation annotations (over the same data). By employing extra training data with bounding box annotations from COCO we are able to match the full supervision results. We also report the first results for weakly supervised instance segmentation, where we also reach $\sim$95% of the quality of the fully-supervised training.

Our current approach exploits existing box-driven segmentation techniques, treating each annotated box individually. In Chapter 5 we consider even a weaker form of supervision for semantic segmentation and propose to train a convnet with image label annotations.

5 EXPLOITING SALIENCY FOR OBJECT SEGMENTATION FROM IMAGE LEVEL LABELS

This chapter studies the problem of training a pixel-wise semantic labeller network from image-level annotations of the present object classes, a much weaker form of supervision compared to Chapters 3 and 4.

Recently, it has been shown that high quality seeds indicating discriminative object regions can be obtained from image-level labels. Without additional information, obtaining the full extent of the object is an inherently ill-posed problem due to co-occurrences. We propose using a saliency model as additional information and hereby exploit prior knowledge on the object extent and image statistics. We show how to combine both information sources in order to recover 80% of the fully supervised performance – which is the new state of the art in weakly supervised training for pixel-wise semantic labelling.

5.1 INTRODUCTION

Semantic image labelling provides a rich information about scenes, but comes at the cost of requiring pixel-wise labelling to generate training data. The accuracy of convnet-based models the correlates strongly with the amount of available training data. Collecting and annotating data has become a bottleneck for progress. This problem has raised interest in exploring partially supervised data or different means of supervision, which represents different tradeoffs between annotation efforts and yield in terms of supervision signal for the learning task. For tasks such as semantic segmentation there is a need to investigate what is the minimal supervision needed to reach quality comparable to the fully supervised case.

A reasonable starting point considers that all training images have image-level labels to indicate the presence or absence of the classes of interest. The weakly supervised learning problem can be seen as a specific instance of learning from constraints (Shcherbatyi and Andres, 2016; Xu *et al.*, 2015). Instead of explicitly supervising the output, the available labels provide a constraint on the desired output. If an image label is absent, no pixel in the image should take that label; if an image label is present at least in one pixel the image must take that label. However, the objects of interest are rarely single pixel. Thus to enforce larger output regions size, shape, or appearance priors are commonly employed (either explicitly or implicitly).

Another reason for exploiting priors, is the fact that the task is fundamentally ambiguous. Strongly co-occurring categories (such as train and rails, sculls and oars, snow-bikes and snow) cannot be separated without additional information. Because

(a)image labels training image (b)saliency (c)our result test image

Figure 5.1: We train a semantic labelling network with (a) image-level labels and (b) saliency masks, to generate (c) a pixel-wise labelling of object classes at test time.

additional information is needed to solve the task, previous work has explored different avenues, including class-specific size priors (Pathak *et al.*, 2015a), crawling additional images (Pinheiro and Collobert, 2015; Wei *et al.*, 2015), or requesting corrections from a human judge (Kolesnikov and Lampert, 2016a; Saleh *et al.*, 2016).

Despite these efforts, the quality of the current best results on the task seems to level out at $\sim$75% of the fully supervised case. Therefore, we argue that additional information sources have to be explored to complement the image level label supervision – in particular addressing the inherent ambiguities of the task. In this work, we propose to exploit class-agnostic saliency as a new ingredient to train for class-specific pixel labelling; and show new state-of-the-art results on Pascal VOC 2012 semantic labelling with image label supervision.

We decompose the problem of object segmentation from image labels into two separate ones: finding the object location (any point on the object), and finding the object's extent. Finding the object extent can be equivalently seen as finding the background area in an image.

For object location we exploit the fact that image classifiers are sensitive to the discriminative areas of an image. Thus training using the image labels enables to find high confidence points over the objects classes of interest (we call these "object seeds"), as well as high confidence regions for background. A classifier, however, will struggle to delineate the fine details of an object instance, since these might not be particularly discriminative.

For finding the object extent, we exploit the fact that a large portion of photos aim at capturing a subject. Using class-agnostic object saliency we can find the segment corresponding to some of the detected object seeds. Albeit saliency is noisy, it provides information delineating the object extent beyond what seeds can indicate. Our experiment show that this is an effective source of additional information. Our saliency model is itself trained from bounding box annotations only. At no point of our pipeline accurate pixel-wise annotations are used.

In this chapter we provide an analysis of the factors that influence the seed generation, explore the utility of saliency for the task, and report best known results both when using image labels only and image labels with additional data.

In summary, our contributions are:

- We propose an effective method for combining seeds and saliency for the task of weakly supervised semantic segmentation. Our method achieves the best performance among the known works that utilise image level supervision with or without additional external data.

- We compare recent seed methods side by side, and analyse the importance of saliency towards the final quality.

Section 5.3 presents our overall architecture, Section 5.4 investigates suitable object seeds, and Section 5.5 describes how we use saliency to guide the convnet training. Finally Section 5.6 discusses the experimental setup, and presents our key results.

5.2 PREVIOUS WORK ON OBJECT LOCALIZATION FROM IMAGE LABELS

Object seeds. Multiple works have considered using a trained classifier (from image level labels) to find areas of the image that belong to a given class, without necessarily enforcing to cover the full object extent (high precision, low recall). Starting from simple strategies such as "probing classifier with different image areas occluded" (Zeiler and Fergus, 2014), or back-propagating the class score gradient on the image (Simonyan *et al.*, 2014); significantly more involved strategies have been proposed, mainly by modifying the back-propagation strategy (Springenberg *et al.*, 2015; Zhang *et al.*, 2016; Shimoda and Yanai, 2016), or by solving a per-image optimization problem (Cao *et al.*, 2015). All these strategies provide some degree of empirical success but lack a clear theoretical justification, and tend to have rather noisy outputs.

Another approach considers modifying the classifier training procedure so as to have it generate object masks as a by-product of a forward-pass. This can be achieved by adding a global max-pooling (Pinheiro and Collobert, 2015) or mean-pooling layer (Zhou *et al.*, 2016) in the last stages of the classifier.

In this work we provide an empirical comparison of existing seeders, and explore variants of the mean-pooling approach (Zhou *et al.*, 2016) (Section 5.4).

Detection boxes from image level supervision. Detecting object boxes from image labels has similar challenges as pixel labelling. The object location and extent need to be found. State-of-the-art techniques for this task (Bilen and Vedaldi, 2016; Teh *et al.*, 2016; Kantorov *et al.*, 2016) learn to re-score detection proposals using two stream architectures that once trained separate "objectness" scores from class scores. These architectures echo with our approach, where the seeds provide information

about the class scores at each pixel (albeit with low recall for foreground classes), and the saliency output provides a per-pixel (class agnostic) "objectness" score.

5.3 GUIDED SEGMENTATION ARCHITECTURE

While previous work has emphasised using sophisticated training losses, or more involved architectures, we focus on saliency as an effective prior, and thus keep our architecture simple.

We approach the image-level supervised semantic segmentation problem via a system with two modules (see Figure 5.2), we name this architecture "Guided Segmentation". Given an image and image-level labels, the "guide labeller" module combines cues from a seeder (Section 5.4) and saliency (Section 5.5) sub-modules, producing a rough segmentation mask (the "guide"). Then a segmenter convnet is trained using the produced guide mask as supervision. In this architecture the segmentation convnet is trained in a fully-supervised procedure, using the traditional per pixel softmax cross-entropy loss.

In sections 5.4 and 5.5 we explain how we build our guide labeller, by first generating seeds (discriminative areas of objects of interest), and then extending them to better cover the full object extents.

5.4 FINDING GOOD SEEDS

There has been recent burst of approaches to localise objects from a classifier. Some approaches rely on image gradients from a trained classifier (Simonyan *et al.*, 2014; Springenberg *et al.*, 2015; Zhang *et al.*, 2016), while others propose to train a global average pooling (GAP) based architectures as a classifier (Zhou *et al.*, 2016). All the classifier based localisation variants have a fundamental limitation in that there exists a mismatch between the training objective (image classification) and the desired output: the object locations. Nonetheless, they have proved to be effective.

In this section, we review the localisation approaches side by side and compare

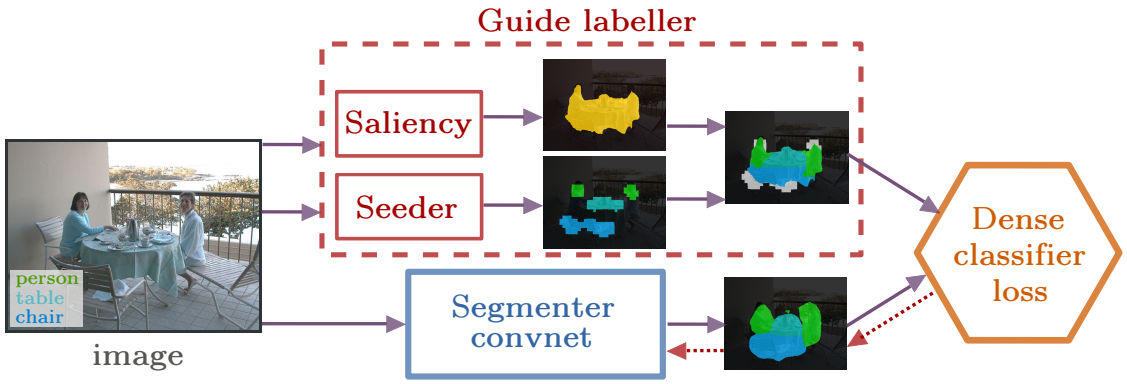


Figure 5.2: High level Guided Segmentation architecture.

GAP	-LowRes (Zhou *et al.*, 2016)	-HighRes (Kolesnikov and Lampert, 2016b)	-ROI	-DeepLab (Chen *et al.*, 2016b)
high res.	✗	✓	✓	✓
dil. conv.	✗	✗	✗	✓
ROI pool	✗	✗	✓	✗
mP	76.5	80.7	80.8	57.7
mAP	88.0	87.0	87.2	92.7

Table 5.1: Architectural comparisons with respect to output resolution, use of dilated convolutions, and region of interest pooling. Mean precision (mP, see text for definition) and classification mean Average Precision (mAP) results are reported.

their empirical performances. We report experimental results of different GAP architectures (Zhou *et al.*, 2016; Kolesnikov and Lampert, 2016b; Chen *et al.*, 2016b), where we show that good architectural components for a classifier or segmenter may not lead to a good GAP architecture.

5.4.1 Global average pooling (GAP)

GAP, or global average pooling layer, can be inserted in the last or penultimate layer of a fully convolutional architecture to turn it into a classifier. The resulting architecture is then trained with a classification loss, and at test time the activation maps before the global average pooling layer have been shown to contain localisation information (Zhou *et al.*, 2016).

In our analysis, we consider four different fully convolutional architectures with a GAP layer: `GAP-LowRes`, `GAP-HighRes`, `GAP-DeepLab`, and `GAP-ROI`. A high-level overview of architectural differences is introduced in Table 5.1. `GAP-LowRes` (Zhou *et al.*, 2016) is essentially a fully convolutional version of VGG-16 (Simonyan and Zisserman, 2015). `GAP-HighRes` is inspired by Kolesnikov and Lampert (2016b) and has 2 times higher output resolution than `GAP-LowRes`. `GAP-DeepLab` is a semantic segmenter DeepLab with a GAP layer over the dense score output. The main difference between `GAP-HighRes` and `GAP-DeepLab` is the presence of dilated convolutions, used to significantly enlarge the field of view in DeepLab. Finally, we consider `GAP-ROI` as a variant of `GAP-HighRes` where we use region of interest pooling to replace sliding window convolutions in the last layers of VGG-16. `GAP-ROI` is meant to be functionally equivalent to `GAP-HighRes`, but with a slight structural variation. As we will see in the next section, this affects GAP's behaviour.

5.4.2 Empirical study

Evaluation. We evaluate each method on the validation set of the Pascal VOC 2012 (Everingham *et al.*) segmentation benchmark. We measure the foreground and background precision-recall curves for each variant. In the foreground case, we

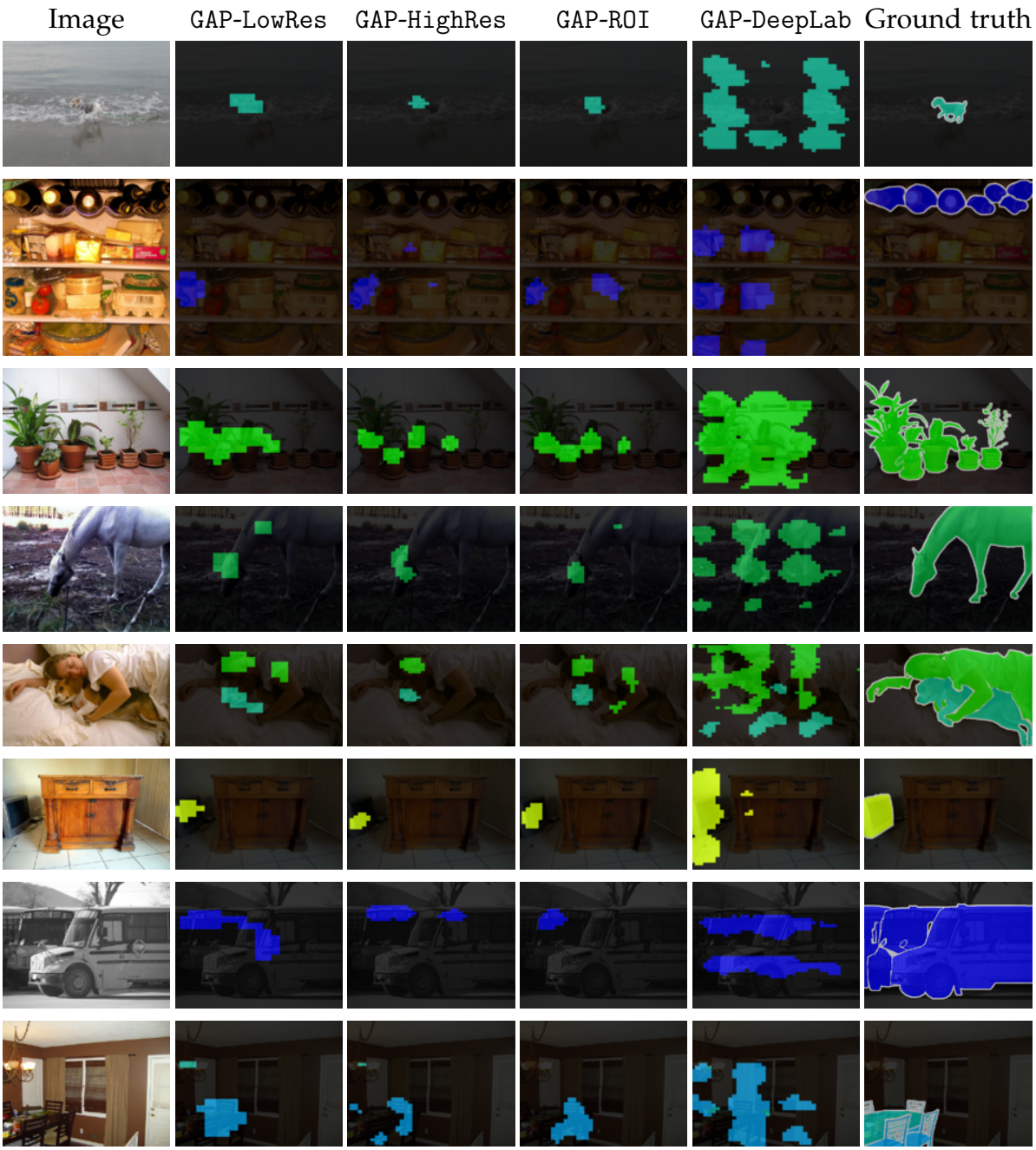
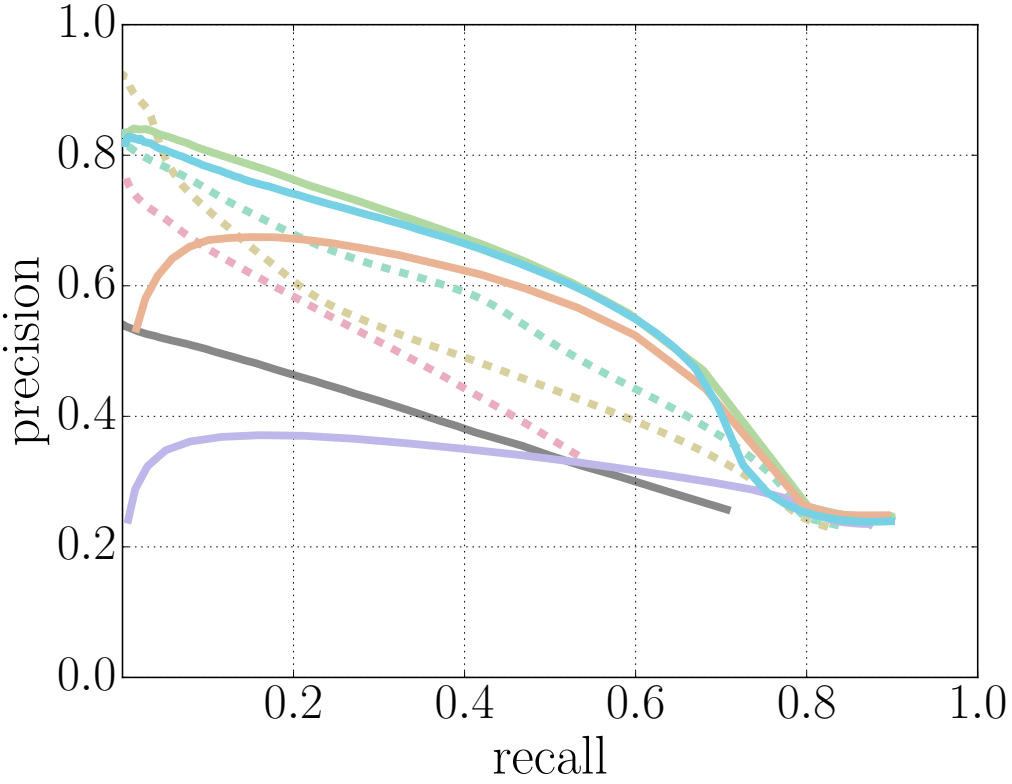
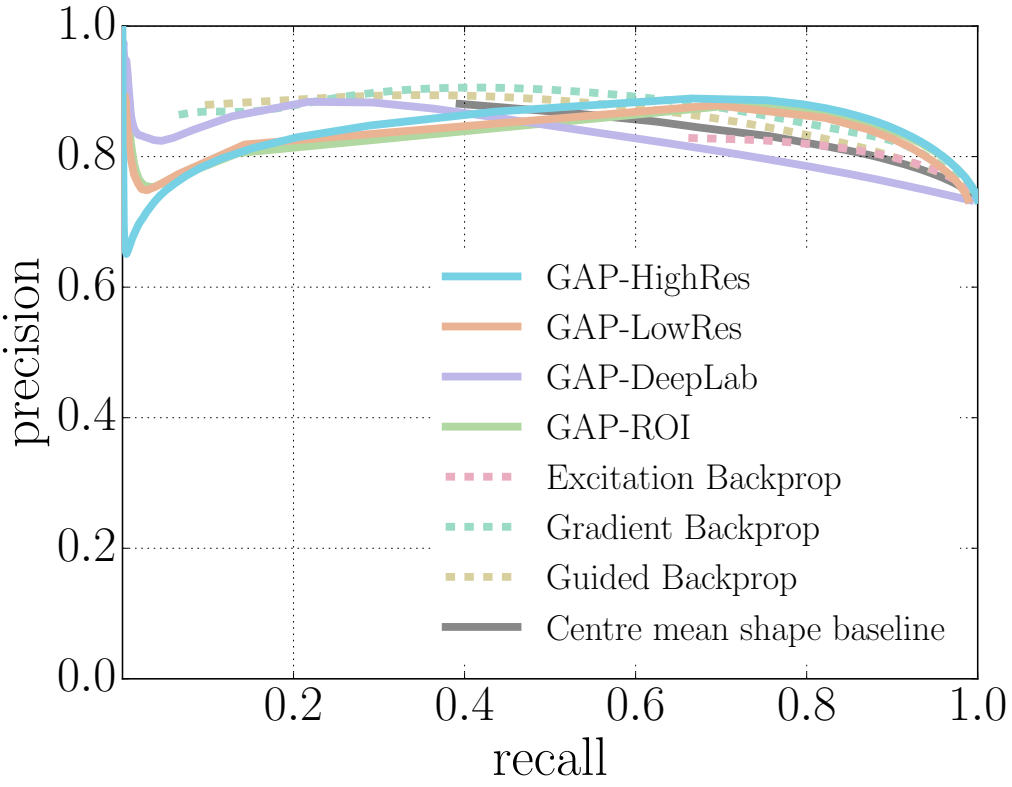


Figure 5.3: Qualitative examples of GAP output for GAP-LowRes, GAP-HighRes, GAP-DeepLab, and GAP-ROI. Note that all of them, except for GAP-DeepLab, are qualitatively similar. For GAP-DeepLab, we observe repeating patterns of certain stride. Examples are chosen at random.



(a) Foreground categories



(b) Background category

Figure 5.4: Comparing seeds techniques. Precision-recall curves.

compute the mean precision and recall over the 20 Pascal categories. The curves are shown in Figure 5.4.

We define mean precision (mP) as a summary metric for the localisation metrics, which averages the foreground precision at 20% recall and the background precision at 80% recall: $\text{mP} = \frac{\text{Prec}_{\text{Fg@20\%}} + \text{Prec}_{\text{Bg@80\%}}}{2}$. Intuitively, for the FG region we only need a small discriminative region, as saliency will fill in the extent. We thus care about precision at ∼20% recall. On the other hand, BG is more diverse and usually takes a larger region; we thus care about precision at ∼80% recall. Since we care about both, we simply take the average (as is the case for the mAP metric). This metric has shown a good correlation with the final performance in our preliminary experiments.

We also measure the classification performance in the standard mean average precision (mAP) metric. Note that seeders are provided with the input image and its ground truth image-level labels.

We compare the GAP architectures against the back-propagation family: Vanilla, Guided, and Excitation back-propagation (Simonyan *et al.*, 2014; Springenberg *et al.*, 2015; Zhang *et al.*, 2016), as well as the centre mean shape baseline, which is a no-image content baseline which predicts an average mask of the all ground truth class instances.

Implementation details. We train all four GAP network variants for multi-label image classification over the *trainaug* set of Pascal VOC 2012. At test time, we take the output per-class heatmaps before the GAP layer and normalise them through dividing by the maximal per-class scores.

For the back-propagation based methods, we use a VGG-16 (Simonyan and Zisserman, 2015) classifier network that has also been trained on the "trainaug" set of Pascal VOC 2012 (10 582 images in total). We take the maximal absolute gradient value among the RGB channels on each pixel as the localisation signal (following Simonyan *et al.* (2014)) and apply Gaussian smoothing. As final post-processing we apply dense CRF (Krähenbühl and Koltun, 2011) to further smooth the seeder output while respecting object boundaries.

In both GAP and backprop variants, we mark as background the pixels where all per-class score values are bellow a given threshold τ, and remaining pixels take the argmax class label.

Results. Refer to Figure 5.4 for the precision-recall curves. GAP variants in general are better localisers than the backprop variants. We note that the Guided backprop gives highest precision at a very low recall regime (∼5%), but we find the recall to be too low to be useful. Among the GAP methods, `GAP-HighRes` and `GAP-ROI` give high precision over most of the recall range. Note that the GAP results depends heavily on the architecture used. For example, `GAP-DeepLab` shows a significantly lower quality than any other GAP variants (despite being the best classifier).

The network matters for GAP. Table 5.1 shows a more detailed view of the GAP results. Despite all architectures being based on VGG-16 the mP results have high

fluctuations (GAP-HighRes: 80.7 mP, GAP-DeepLab: 57.7 mP), while there is no such dramatic effect in the performance as classifiers (mAP). It is striking that GAP-DeepLab is the best classifier, while giving the lowest performance in localisation when trained with GAP. Thus better classifiers (even based on a semantic labelling network) do not automatically make better seeders.

Along the architectural component dimensions, we observe that a higher resolution network performs better as a seeder than their lower resolution counterpart (GAP-HighRes versus GAP-LowRes), while using a larger field of view through dilated convolutions hurts the GAP performance (GAP-HighRes versus GAP-DeepLab). We observe on-par performances between GAP-HighRes and GAP-ROI.

In the rest of the chapter, we use GAP-HighRes as the seeder module. In Kolesnikov and Lampert (2016b), foreground and background seeds are handled via two different mechanisms, in our experiments we simply treat all the non-foreground region as background.

5.5 FINDING THE OBJECT EXTENT

Having generated a set of seeds indicating discriminative object areas, the guide labeller needs to find the extent of the object instances (Section 5.3).

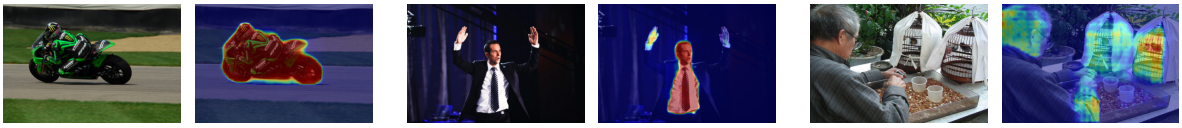
Without any prior knowledge, it is very hard, if not impossible, to learn the extent of objects only from images and image-level labels only. Image-level labels only convey information about commonly occurring patterns that are present in images with positive tags and absent in images with negative tags. The system is thus susceptible to strong inter-class co-occurrences (e.g. train with rail), as well as systematic part occlusions (e.g. feet).

CRF and CRFLoss. A traditional approach to make labels match object boundaries is to solve a CRF inference problem (Lafferty *et al.*, 2001; Krähenbühl and Koltun, 2011) over the image grid, where pair-wise terms relate to the object boundaries. A CRF can be applied at three stages: (1) on the seeds (crf-seed), (2) as a loss function during segmenter convnet training (crf-loss) (Kolesnikov and Lampert, 2016b), and (3) as a post-processing at test time (crf-postproc).

We have experimented with multiple combinations of those. Albeit some gains are observed, these are inconsistent. For example GAP-HighRes and GAP-ROI provide near identical classification and seeding performance (see Table 5.1), yet using the same CRF setup will provide +13 mIoU percent points in one, but only +7 pp on the other. In comparison our saliency approach will provide +17 mIoU and +18 mIoU for these two networks respectively (see below).

5.5.1 Saliency

Image saliency has multiple connotations: it can refer to a spatial probability map of where a person might look first (Yamada *et al.*, 2010), a probability map of which



(a) High quality (b) Medium quality (c) Low quality

Figure 5.5: Example of our saliency map results on Pascal VOC 2012 data.

object a person might look at first (Li *et al.*, 2014), or a binary mask segmenting the one object a person is most likely to look first (Borji *et al.*, 2015; Shi *et al.*, 2016). We employ the latter definition in this work. Note that this notion is class-agnostic, and refers more to the composition of the image, than the specific object category.

In this chapter we propose to use object saliency to extract information about the object extent. We work under the assumption that a large portion of the dataset is intentional photographies, which is the case for most datasets crawled from the web such as Pascal (Everingham *et al.*) and COCO (Lin *et al.*, 2014). If the image contains a single label "dog", chances are that the image is about a dog, and that the salient object of the image is a dog. We use a convnet based saliency estimator (detailed in Section 5.6.1) which adds the benefit of translation invariance. If two locally salient dogs appear in the image, both will be labelled as foreground.

When using saliency to guide semantic labelling at least two difficulties need to be handled. For one, saliency per-se does not segment object instances. In the example Figure 5.5a, the person-bike is well segmented, but person and bike are not separated. Yet the ideal Guide labeller (Figure 5.2) should give different labels to these two objects. The second difficulty, clearly visible in the examples of Figure 5.5, is that the salient object might not belong to a category of interest (shirt instead of person in Figure 5.5b) or that the method fails to identify any salient region at all (Figure 5.5c).

We measure the saliency quality when compared to the ground truth foreground on the Pascal VOC 2012 validation set. Albeit our convnet saliency model is better than hand-crafted methods (Jiang *et al.*, 2013; Zhang *et al.*, 2015a), in the end only about 20% of images have reasonably good (IoU > 0.6) foreground saliency quality. Yet, as we will see in Section 5.6, this bit of information is already helpful for the weakly supervised learning task.

Crucially, our saliency system is trained on images containing diverse objects (hundreds of categories), the object categories are treated as "unknown", and to ensure clean experiments we handicap the system by removing any instance of Pascal categories in the object saliency training set. Our saliency model captures a general notion of plausible foreground objects and background areas (more details in Section 5.6.1).

On every Pascal training image, we obtain a class-agnostic foreground/background binary mask from our saliency model, and high precision/low recall class-specific image labels from the seeds model (Section 5.4). We want to combine them in such a way that seed signals are well propagated throughout the foreground saliency mask.

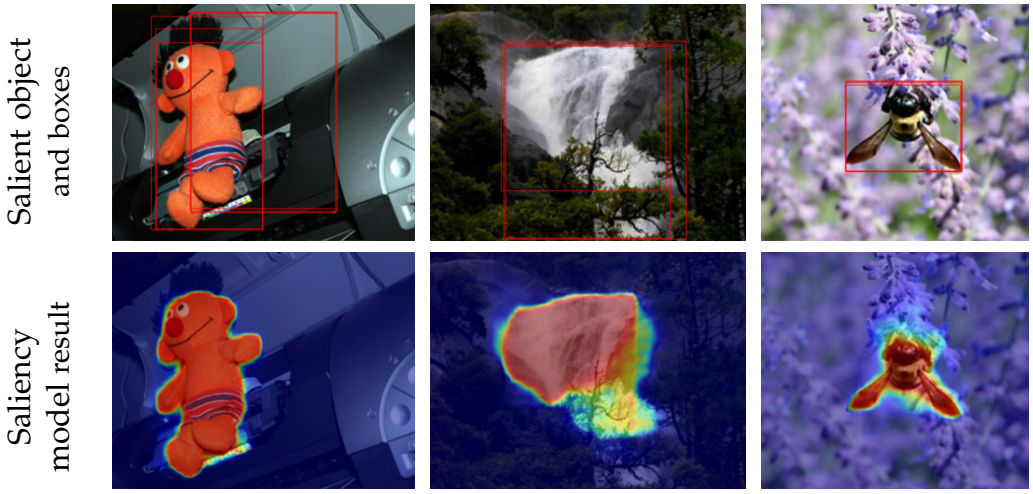


Figure 5.6: Example of saliency results on its training data. We use MSRA box annotations to train a weakly supervised saliency model. Note that the MSRA subset employed does not contain Pascal categories.

We consider two baselines strategies to generate guide labels using saliency but no seeds ($\mathcal{G}_0$ and $\mathcal{G}_1$), and then discuss how we combine saliency with seeds ($\mathcal{G}_2$).

$\mathcal{G}_0$ **Random class assignment.** Given a saliency mask, we assign all foreground pixels to a class randomly picked from the ground truth image labels. If a single "dog" label is present, then all foreground pixels are "dog". Two labels are present ("dog, cat"), then all pixels are either dog or cat.

$\mathcal{G}_1$ **Per-connected component classification.** Given a saliency mask, we split it in components, and assign a separate label for each component. The per-component labels are given using a full-image classifier trained using the image labels (classifier details in Section 5.6.1). Given a connected component mask R_i^{fg} (with pixel values 1: foreground, 0: background), we compute the classifier scores when feeding the original image (I), and when feeding an image with background zeroed ($I \odot R_i^{fg}$). Region R_i^{fg} will be labelled with the ground truth class with the greatest positive score difference before and after zeroing.

$\mathcal{G}_2$ **Propagating seeds.** Here, instead of assigning the label per connected component R_i^{fg} using a classifier, we instead use the seed labels. We also treat the seeds as a set of connected components (seed R_j^s). Depending on how the seeds and the foreground regions intersect, we decide the label for each pixel in the guide labeller output.

Our fusion strategy uses five simple ideas. 1) We treat the seeds as reliable small size point predictors of each object instance, but that might leak outside of the object. 2) We assume the saliency might trigger on objects that are not part of the classes of

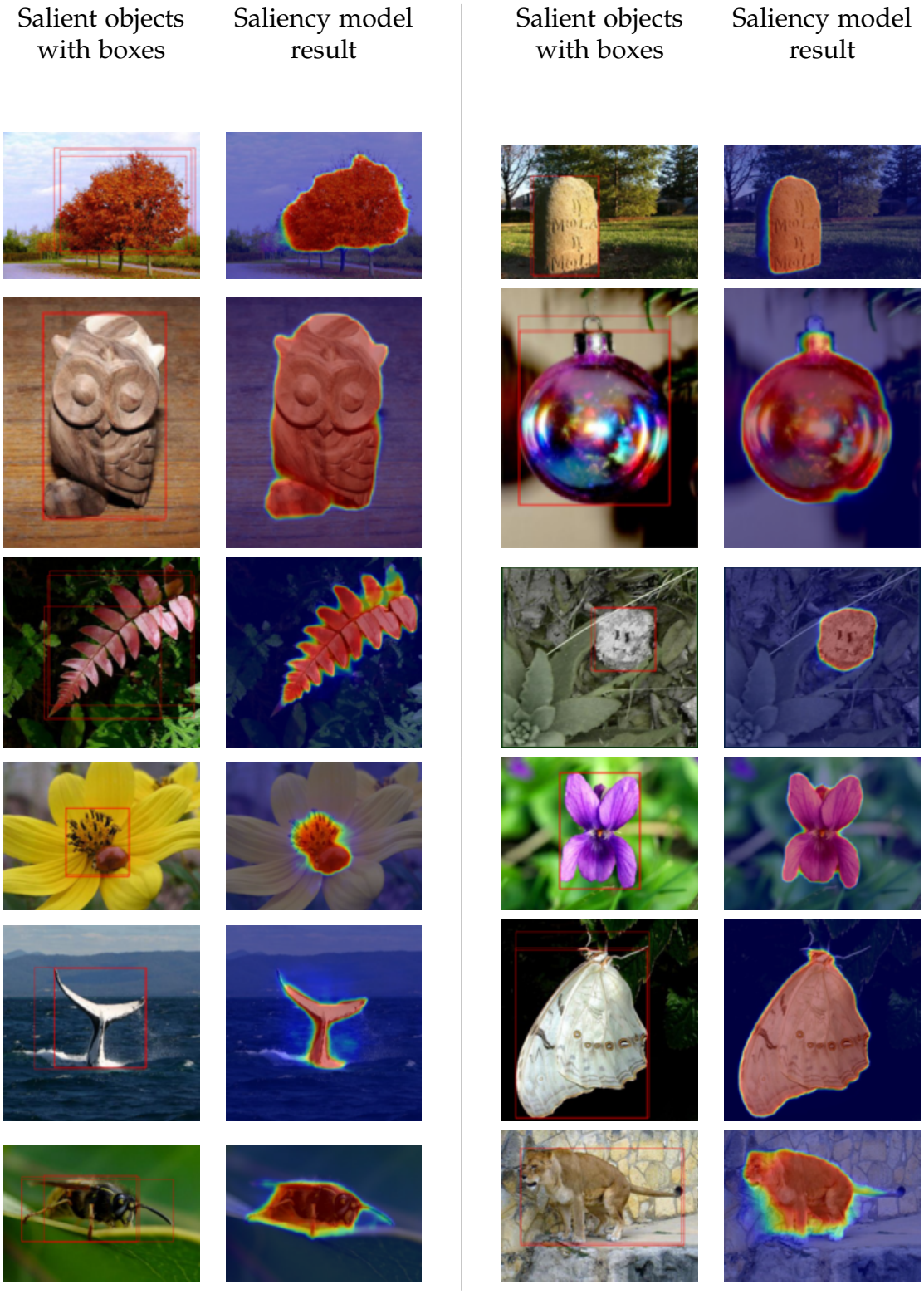


Figure 5.7: Extension of Figure 5.6. Examples of saliency results on its training data. We use MSRA box annotations to train a weakly supervised saliency model. Note that the MSRA subset employed is not biased towards the Pascal categories. Examples are chosen at random.

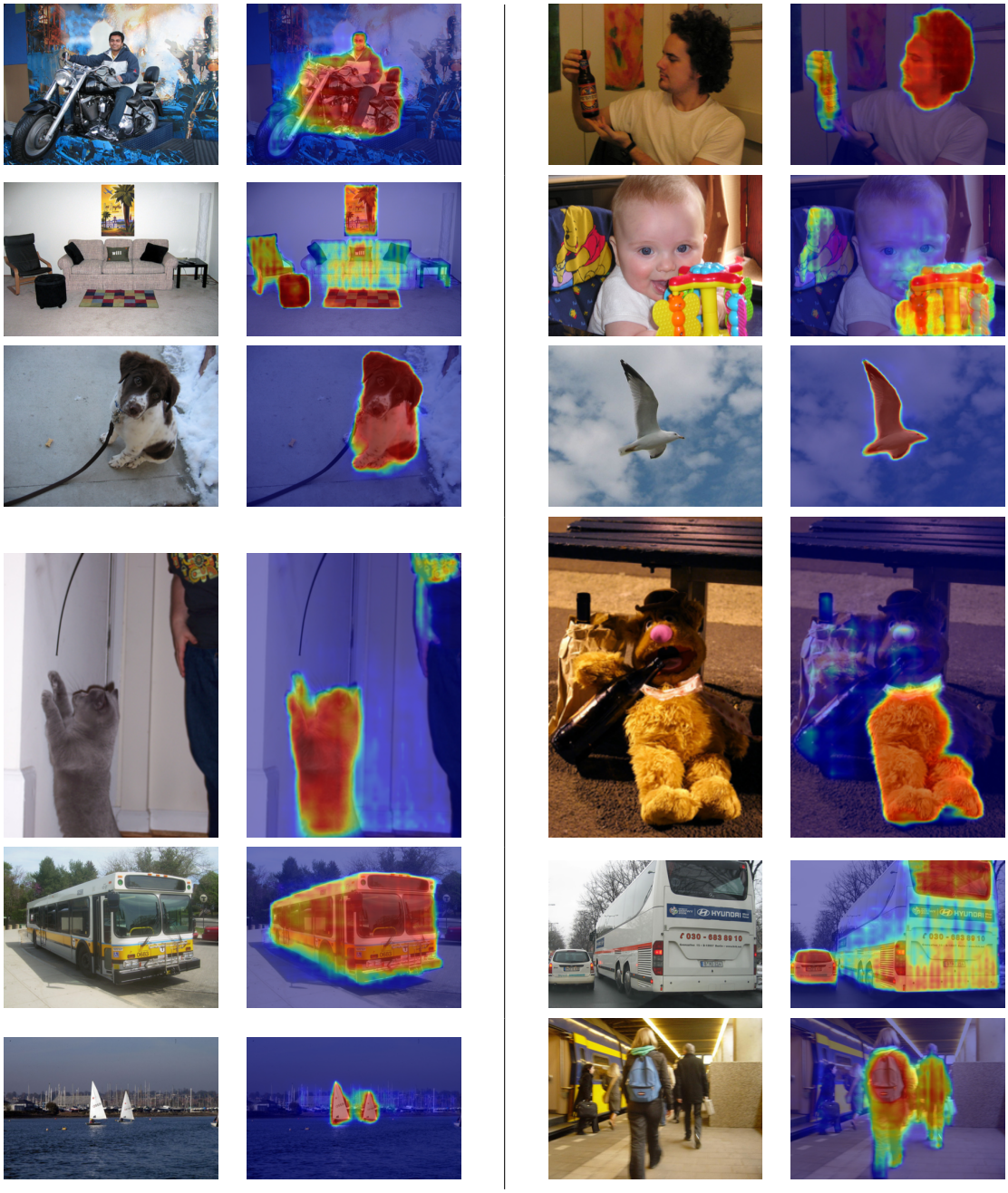
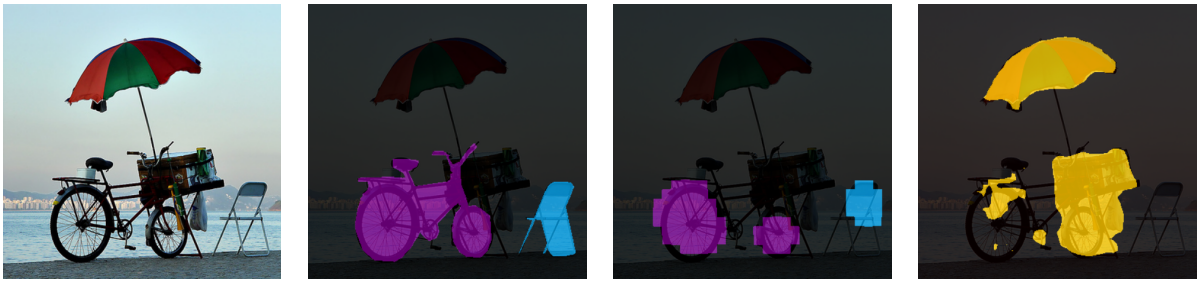


Figure 5.8: Extension of Figure 5.5. Example of saliency results on Pascal images. We note that the saliency often fails when the central, salient objects are non-Pascal or when the scene is cluttered. Examples are chosen at random.



(a) Image (b) Ground truth (c) Seed (d) Saliency

(e) $\mathcal{G}_0$ (f) $\mathcal{G}_1$ (g) $\mathcal{G}_2$

Figure 5.9: Guide labelling strategies example results. The image, its labels ("bicycle, chair"), seeds, and saliency map are their input. White overlay indicates "ignore" pixel label.

interest. 3) A foreground connected component R_i^{fg} should take the label of the seed touching it. 4) If two (or more) seeds touch the same foreground component, then we want to propagate all the seed labels inside it. 5) When in doubt, mark as ignore.

Figure 5.9 provides example results of the different guide strategies. For additional qualitative examples of seeds, saliency foreground, and generated labels, see Figure 5.11. With our guide strategies $\mathcal{G}_0$, $\mathcal{G}_1$, and $\mathcal{G}_2$ at hand, we now proceed to empirically evaluate them in Section 5.6.

5.6 EXPERIMENTS

Sections 5.6 and 5.6.1 provide the details of the evaluation and our implementation. Section 5.6.2 compares our different guide strategies amongst each other, and Section 5.6.3 compares with previous work on weakly supervised semantic labelling from image-level labels.

Evaluation. We evaluate our image-level supervised semantic segmentation system on the PASCAL VOC 2012 segmentation benchmark (Everingham *et al.*). We report all the intermediate results on the validation set (1 449 images) and only report the final system result on the test set (1 456 images). Evaluation metric is the standard mean intersection-over-union (mIoU) measure.

5.6.1 Implementation details

For training "Seeder" and "Segmenter" networks, we use the ImageNet (Deng *et al.*, 2009) pretrained models for initialisation and fine-tune on the Pascal VOC 2012 *trainaug* set (10 582 images), an extension of the original train set (1 464 images) (Everingham *et al.*; Hariharan *et al.*, 2011). This is the same procedure used by previous work on fully (Chen *et al.*, 2016b) and weakly supervised learning (Kolesnikov and Lampert, 2016b).

Seeder. Results in Tables 5.2 and 5.3 are obtained using `GAP-HighRes` (see Section 5.4), trained for image classification on the Pascal trainaug set. The test time foreground threshold τ is set to 0.2, following the previous literature (Zhou *et al.*, 2016; Kolesnikov and Lampert, 2016b).

$\mathcal{G}_1$ **Classifier.** The guide labeller strategy $\mathcal{G}_1$ uses an image classifier trained on Pascal trainaug set. We use the VGG-16 architecture (Simonyan and Zisserman, 2015) with a multi-label loss.

Saliency. Following Zhao *et al.* (2015); Li *et al.* (2016b); Li and Yu (2016) we repurpose a semantic labelling network for the task of class-agnostic saliency. We train a DeepLab-v2 ResNet network (Chen *et al.*, 2016b) over a subset of MSRA (Liu *et al.*, 2011), a saliency dataset with *class agnostic* bounding box annotations. We constrain the training only to data samples of *non-Pascal* categories. Thus, the saliency model does not leverage class specific features when Pascal images are fed. Out of 25k MSRA images, 11 041 are selected after filtering.

MRSA provides bounding boxes (from multiple annotators) of the main salient element of each image. To train the saliency model to output pixel-wise masks, we follow the approach proposed in Chapter 4 (Khoreva *et al.*, 2017a). We generate segments from the MSRA boxes by applying grabcut over the average box annotation, and use these as supervision for the DeepLab model. The model is trained as a binary semantic labeller for foreground and background regions. The trained model generates masks like the ones shown in Figure 5.6. Although having been trained with images with single salient objects, due to its convolutional nature the network can predict multiple salient regions in the Pascal images (as shown in Figure 5.11).

At test time, the saliency model generates a heatmap of foreground probabilities. We take pixels with $\geq 50\%$ of the maximal foreground probability as our saliency foreground mask.

Segmenter. For comparison with previous work we use the DeepLabv1-LargeFOV (Chen *et al.*, 2016b) architecture as our segmenter convnet. The network is trained on Pascal trainaug set with 10 582 images, using the output of the guide labeller (Section 5.2), which uses only the image and presence-absence tags of the 20 Pascal categories as supervision. The network is trained for 8k iterations.

Following the standard DeepLab procedure, at test time we up-sample the output

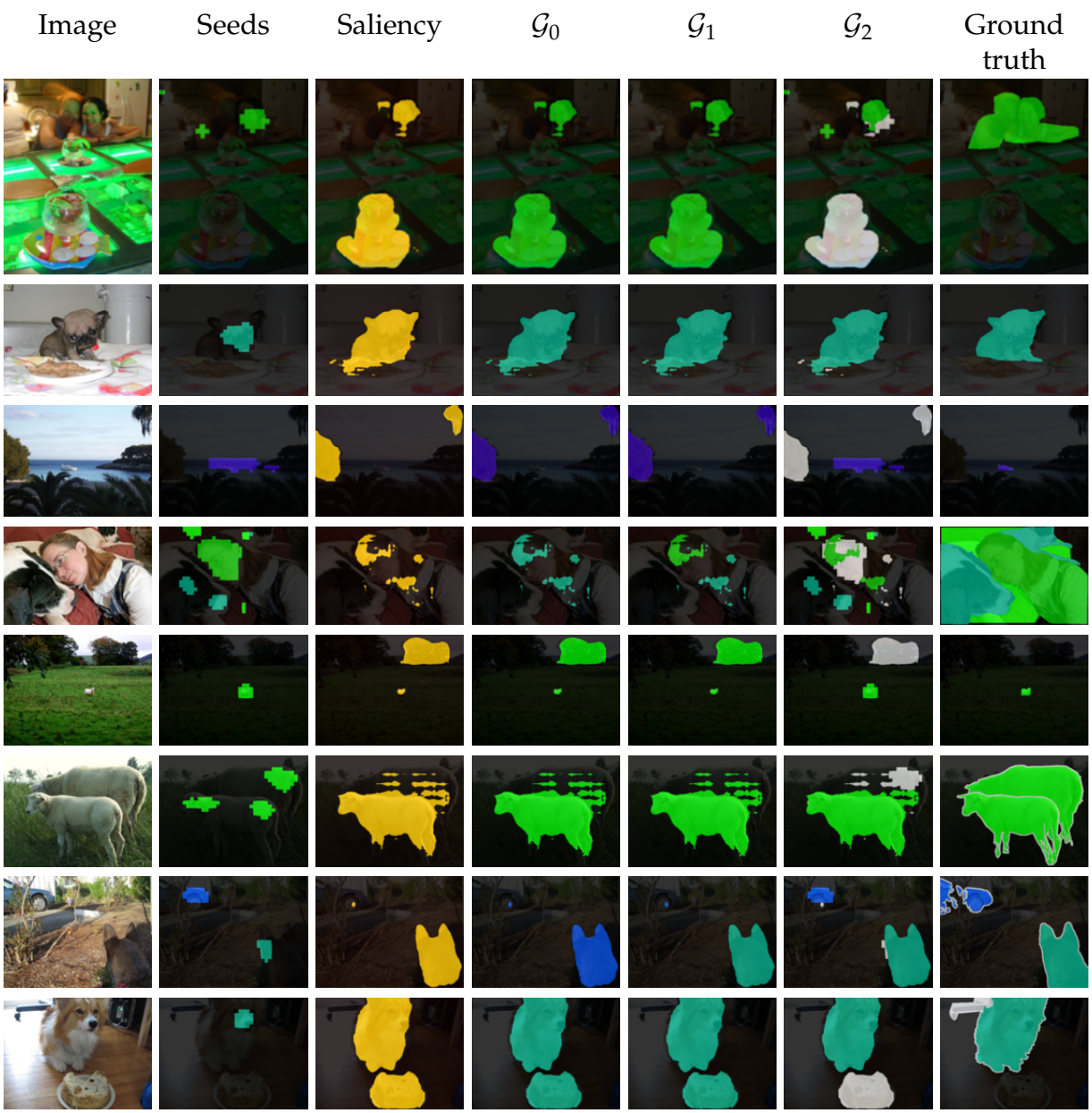


Figure 5.10: Extension of Figure 5.9. Example results for three different guide labelling strategies, $\mathcal{G}_0$, $\mathcal{G}_1$, and $\mathcal{G}_2$. The image, its image labels, seeds, and saliency map are their input. White labels indicate "ignore" regions. Note that $\mathcal{G}_0$ and $\mathcal{G}_1$ give qualitatively similar results, while $\mathcal{G}_2$ produces much more precise labelling by exploiting rich localisation information from the seeds. Examples are chosen at random.

Method	Seeds	Sali-ency	Supervision Fg P/R		Bg P/R		val. set mIoU
Seeds only	✓	✗	69	37	81	95	38.7
$\mathcal{G}_0$	✗	✓	65	52	65	52	45.8
$\mathcal{G}_1$	✗	✓	75	51	75	51	46.2
$\mathcal{G}_2$	✓	✓	73	59	87	95	51.2
Saliency oracle	✓	✓	89	91	100	99	56.9

Table 5.2: Comparison of different guide labeller variants. Pascal VOC 2012 validation set results, without CRF post-processing. Fg/Bg P/R: are foreground/background precision and recall of the guide labels. Discussion in Section 5.6.2.

to the original image resolution and apply the dense CRF inference (Krähenbühl and Koltun, 2011). Unless stated otherwise, we use the CRF parameters used for DeepLabv1-LargeFOV (Chen *et al.*, 2016b).

5.6.2 Ingredients study

Table 5.2 compares different guide strategies $\mathcal{G}_0$, $\mathcal{G}_1$, $\mathcal{G}_2$, and oracle versions of $\mathcal{G}_2$. The first row shows the result of training our segmenter using the seeds directly as guide labels. This leads to poor quality (38.7 mIoU). The "Supervision" column shows recall and precision for foreground and background of the guide labels themselves (training data for the segmenter). We can see that the seeds alone have low recall for the foreground (37%). In comparison, using saliency only, $\mathcal{G}_0$ reaches significantly better results, due to the guide labels having higher foreground recall (52%, while keeping a comparable precision).

Adding a classifier on top of the saliency ($\mathcal{G}_0 \rightarrow \mathcal{G}_1$) provides only a negligible improvement (45.8 $\rightarrow$ 46.2). This can be attributed to the fact that many Pascal images contain only a single foreground class, and that the classifier might have difficulties recognizing the masked objects. Interestingly, when using a similar classifier to generate seeds instead of scoring the image ($\mathcal{G}_1 \rightarrow \mathcal{G}_2$) we gain 5 pp (percent points, 46.2 $\rightarrow$ 51.2). This shows that the details of how a classifier is used can make a large difference.

Table 5.2 also reports a saliency oracle case on top of $\mathcal{G}_2$. If we use the ground truth annotation to generate an ideal saliency mask we see a significant improvement over $\mathcal{G}_2$ (51.2 $\rightarrow$ 56.9). This shows that the quality of the saliency is an important ingredient, and that there is room for further gains.

5.6.3 Results

Table 5.3 compares our results with previous related work. We group results by methods that only use ImageNet pre-training and image-level labels (I, P, E; see

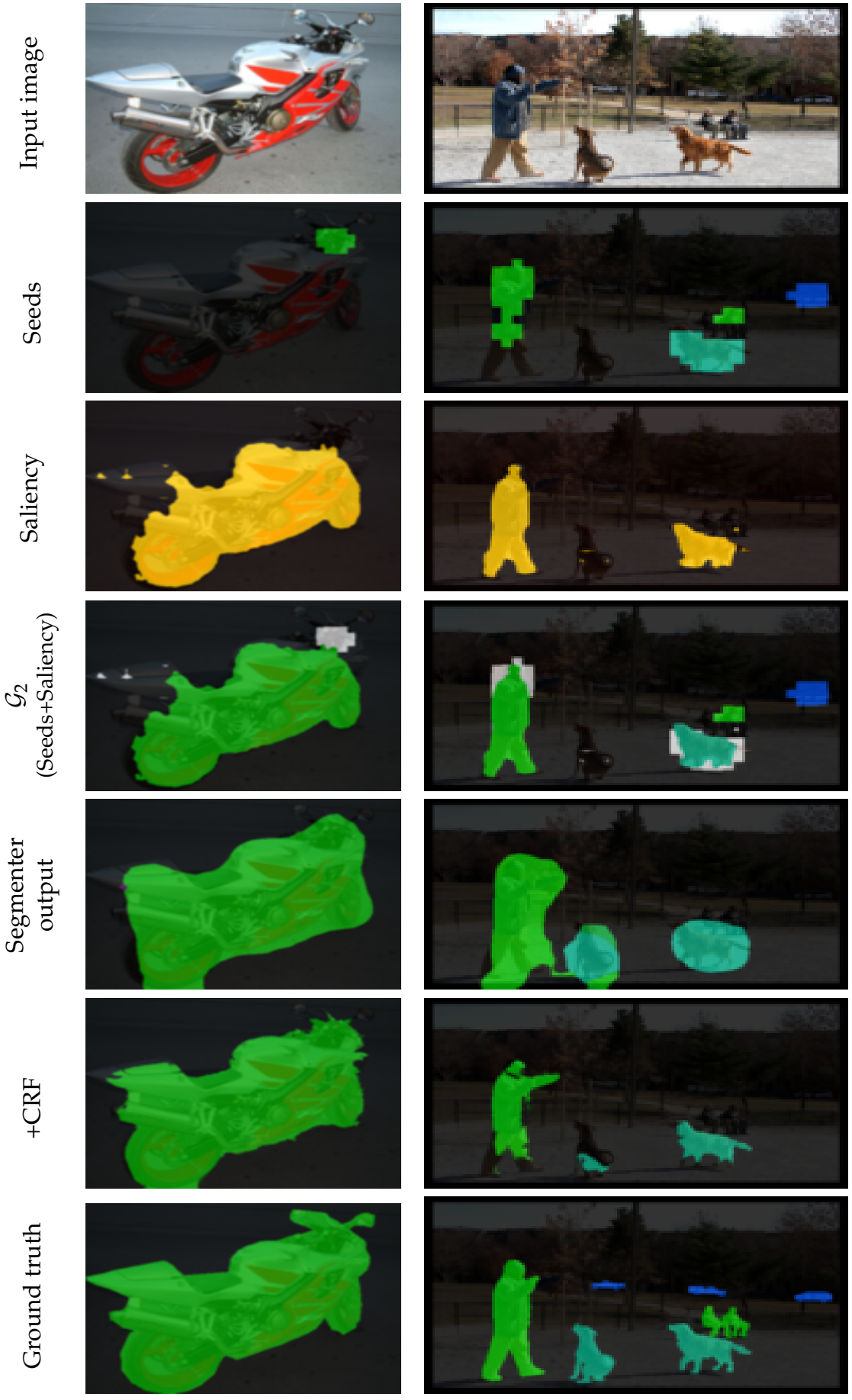


Figure 5.11: Qualitative examples of the different stages of our system.

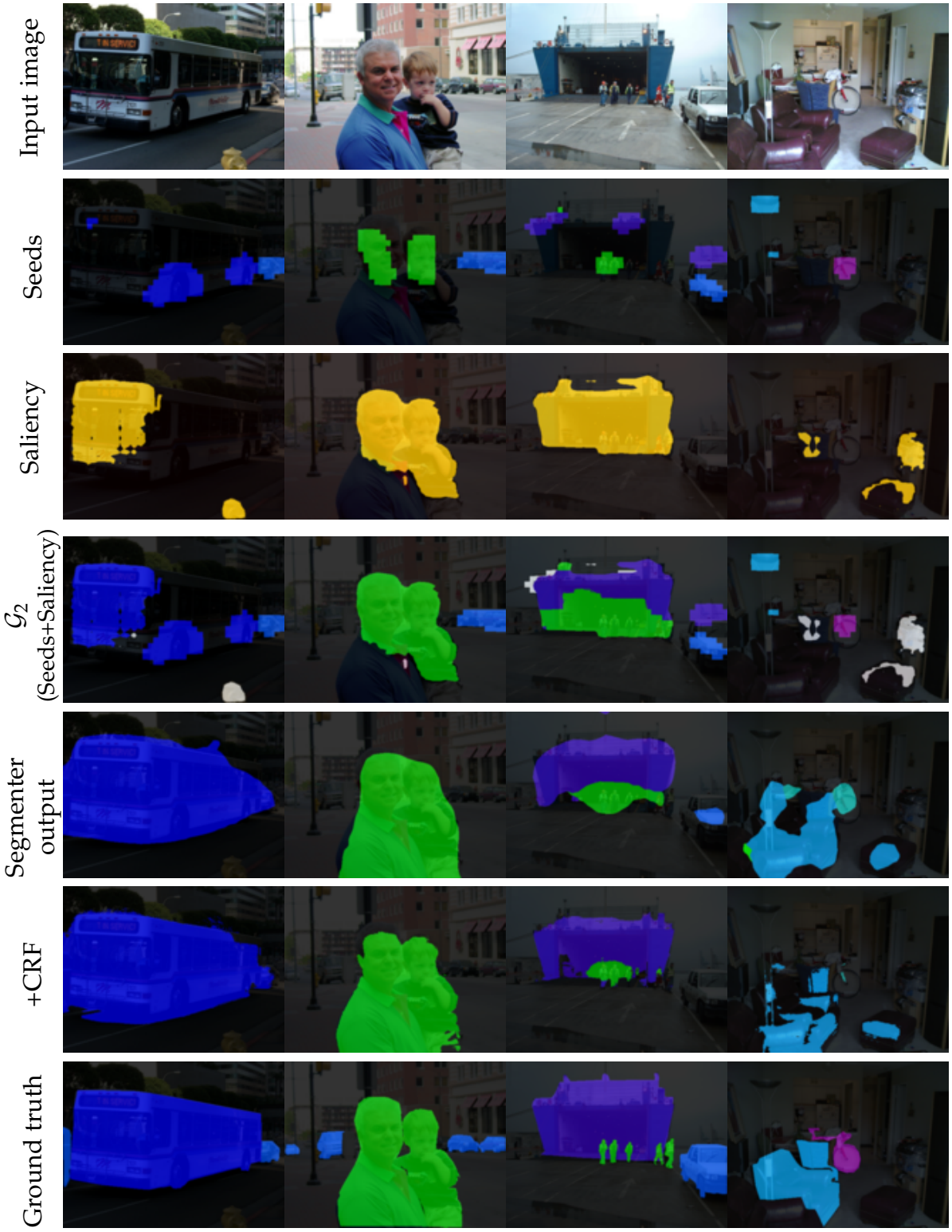


Figure 5.12: More qualitative examples of the different stages of the Guided Segmentation system on the training images. White labels are "ignore" regions. Seeds have high precision and low recall; combined with saliency foreground mask using $\mathcal{G}_2$ guide labeller, object extents are recovered. The generated guide labelling can still be noisy; however, the segmenter convnet can average out the noise to produce more precise predictions. CRF post-processing further refines the predictions.

	Method	Data	val. set mIoU	test set mIoU	test set FS%
Image labels only	`MIL-FCN` (Pathak *et al.*, 2015b)	I+P	25.0	25.6	36.5
	`CCNN` (Pathak *et al.*, 2015a)	I+P	35.3	35.6	50.6
	`WSSL` (Papandreou *et al.*, 2015)	I+P	38.2	39.6	56.3
	`MIL+Seg` (Pinheiro and Collobert, 2015)	I+E_{760k}	42.0	40.6	57.8
	`DCSM` (Shimoda and Yanai, 2016)	I+P	44.1	45.1	64.2
	`CheckMask` (Saleh *et al.*, 2016)	I+P	46.6	-	-
	`SEC` (Kolesnikov and Lampert, 2016b)	I+P	50.7	51.7	73.5
	`AF-ss` (Qi *et al.*, 2016)	I+P	51.6	-	-
	Seeds only	I+P	39.8	-	-
More information	`CCNN` (Pathak *et al.*, 2015a)	I+P+Z	-	45.1	64.2
	`STC` (Wei *et al.*, 2015)	I+P+S+E_{40k}	49.8	51.2	72.8
	`CheckMask` (Saleh *et al.*, 2016)	I+P+μ	51.5	-	-
	`MicroAnno` (Kolesnikov and Lampert, 2016a)	I+P+μ	51.9	53.2	75.7
	$\mathcal{G}_0$	I+P+S	48.8	-	-
	$\mathcal{G}_2$	I+P+S	**55.7**	**56.7**	**80.6**
	DeepLabv1	I+P_{full}	67.6	70.3	100

Table 5.3: Comparison of state-of-the-art methods, on Pascal VOC 2012 val. and test set. FS%: fully supervised percent. Ingredients: I: ImageNet classification pre-training, P: Pascal image level tags, P_{full}: fully supervised case (pixel wise labels), E_n: n extra images with image level tags, S: saliency, Z: per-class size prior, μ: human-in-the-loop micro-annotations.

legend Table 5.3), and methods that use additional data or user-inputs. Here our $\mathcal{G}_0$ and $\mathcal{G}_2$ results include a CRF post-processing (`crf-postproc`). We also experimented with `crf-loss` but did not find a parameter set that provided improved results.

We see that the guide strategies $\mathcal{G}_0$, which uses saliency and random ground-truth label, reaches competitive performance compared to methods using I+P only. This shows that saliency by itself is already a strong cue. Our guide strategy $\mathcal{G}_2$ (which uses seeds and saliency) obtains the best reported results on this task[1]. We even improve over other methods using saliency (`STC`) or using additional human annotations (`MicroAnno`, `CheckMask`). Compared to a fully supervised DeepLabv1 model, our results reach 80% of the fully supervised quality.

[1]Qi *et al.* (2016) also report 54.3 validation set results; however, we do not consider these results comparable since they use the MCG scores (Pont-Tuset *et al.*, 2016), which are trained on the ground truth Pascal segments.

5.7 CONCLUSION

We have addressed the problem of training a semantic segmentation convnet from image labels. Image labels alone can provide high quality seeds, or discriminative object regions, but learning the full object extents is a hard problem. We have shown that saliency is a viable option for obtaining the object extent information.

The proposed Guided Segmentation architecture (Section 5.3), where the "guide labeller" combines cues from the seeds and saliency, can successfully train a segmentation convnet to achieve state-of-the-art performance. Our weakly supervised results reach 80% of the fully supervised case.

We expect that a deeper understanding of the seeder methods and improvements on the saliency model can lead to further improvements.

Part II

LEARNING TO SEGMENT VIDEOS VIA GRAPHS

A popular and successful approach is modeling video segmentation as a graph partitioning problem, where the nodes represent pixels or superpixels, and the edges encode the spatio-temporal structure. These methods usually consist of three essential steps: 1. extraction of superpixels and feature computation; 2. graph construction; 3. partitioning of the graph using spectral clustering. In this part of the thesis we present our proposed improvements for each individual step.

In Chapter 6 we address step 3 and propose to integrate the learned must-links constraints into spectral clustering framework in order to reduce the computational load as well as to guide the segmentation towards the right solution. In Chapter 7 we focus on step 2 and explore how to construct a graph to obtain the best video segmentation performance. We propose to learn the topology of the graph and its edge weights from the features estimated in step 1. Learning the graph helps to improve the results, while significantly reducing its runtime, as the learnt graph is much sparser. Chapter 8 addresses step 1 and proposes better superpixels for video segmentation. We show that boundary-based superpixels perform best, and that boundary estimation can be improved by fusion of appearance and motion cues. By employing as graph nodes superpixels generated from better boundaries we observe consistent improvement.

6 LEARNING MUST-LINK CONSTRAINTS FOR VIDEO SEGMENTATION BASED ON SPECTRAL CLUSTERING

In recent years it has been shown that clustering and segmentation methods can greatly benefit from the integration of prior information in terms of must-link constraints. Very recently the use of such constraints has been integrated in a rigorous manner also in graph-based methods such as normalized cut. On the other hand spectral clustering as relaxation of the normalized cut has been shown to be among the best methods for video segmentation.

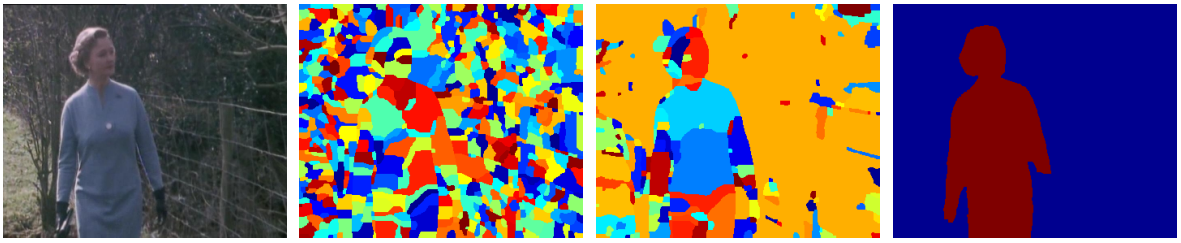
In this chapter we merge these two developments and propose to learn must-link constraints for video segmentation with spectral clustering. We show that the integration of learned must-link constraints not only improves the segmentation result but also significantly reduces the required runtime, making the use of costly spectral methods possible for today's high quality video.

6.1 INTRODUCTION

Video segmentation is an open problem in computer vision, which has recently attracted increasing attention. The problem is of high interest due to its potential applications in action recognition, scene classification, 3D reconstruction and video indexing, among others. The literature on the topic has become prolific (Brendel and Todorovic, 2009; Vazquez-Reina *et al.*, 2010; Andres *et al.*, 2011; Lezama *et al.*, 2011; Cheng and Ahuja, 2012; Chang *et al.*, 2013; Banica *et al.*, 2013; Li *et al.*, 2013) and a number of techniques have become available, e.g. generative layered models (Kannan *et al.*, 2005; Kumar *et al.*, 2008), graph-based models (Grundmann *et al.*, 2010; Xu and Corso, 2012; Palou and Salembier, 2013) and spectral techniques (Shi and Malik, 2000; Brox and Malik, 2010; Fragkiadaki and Shi, 2012; Galasso *et al.*, 2012; Maire and Yu, 2013; Ochs *et al.*, 2014; Galasso *et al.*, 2014).

Spectral methods, stemming from the seminal work of Shi and Malik (2000) and Ng *et al.* (2001), have received much attention from the theoretical viewpoint (von Luxburg, 2007; Bühler and Hein, 2009; Hein and Bühler, 2010), and have proven to be successful for segmentation (Arbeláez *et al.*, 2011; Sundaram and Keutzer, 2011; Galasso *et al.*, 2012; Sundberg *et al.*, 2011; Ochs *et al.*, 2014; Taylor, 2013; Maire and Yu, 2013; Galasso *et al.*, 2014). Spectral clustering, as a relaxation of the NP-hard normalized cut problem, is suitable due to its ability to include long-range affinities (Galasso *et al.*, 2012; Sundaram and Keutzer, 2011) and its global view on the problem (Fowlkes and Malik, 2004), providing balanced solutions.

In this chapter, we focus on two important limitations of spectral techniques: the *excessive resource requirements* and the *lack of exploiting available training data*. The



a) Video sequence b) SPX c) Proposed $\mathcal{M}$ SPX d) Video segm.

Figure 6.1: Video segmentation (Galasso *et al.*, 2012) employs fine superpixels (*b*), resulting in large resource requirements, *esp.* when using spectral methods. We propose learned must-links to merge superpixels into fewer must-link-constrained $\mathcal{M}$ superpixels (*c*). This reduces runtime and memory consumption and maintains or improves the segmentation (*d*).

large demands of spectral techniques (Sundaram and Keutzer, 2011; Galasso *et al.*, 2012) are particularly clear in the case of high-quality video datasets (Galasso *et al.*, 2013), limiting their current large-scale applicability. While often a labeled dataset is available, a systematic learning of the affinities used to build the graph for spectral clustering is very difficult. In particular, as the normalized cut itself is a NP-hard problem and even the spectral relaxation is non-convex, the optimization of the minimizer which yields the segmentation is out of reach. Thus in practice one typically validates a few model parameters (Brox and Malik, 2010; Galasso *et al.*, 2012; Maire and Yu, 2013), preventing spectral methods to make use of recently available large training data (Galasso *et al.*, 2013).

We propose to *learn must-link constraints* to overcome both limitations. Recent spectral theory work (Rangapuram and Hein, 2012; Galasso *et al.*, 2014) has shown that the integration of must-links (i.e. forcing two vertices to be in the same cluster) allows to reduce the size of the problem, while preserving the original optimization objective for all partitions satisfying the must-links. On the other hand by learning must-link constraints we can leverage the available training data in order to guide spectral clustering towards a desired segmentation. Figure 6.1 illustrates the advantages of learning must-links: superpixel-based techniques (Galasso *et al.*, 2012) build spectral graphs on fine superpixels, Figure 6.1(b); by contrast, we propose to build graphs merging superpixels based on learned must-link constraints, Figure 6.1(c). In particular, specifically training a classifier to minimize the number of false positives allows conservative superpixel merging, which: 1. reduces the problem size significantly; 2. preserves the original optimization problem; and 3. improves the video segmentation, Figure 6.1(d), because *correct* must-links avoid undesired solutions (cf. Section 6.3).

In the following, we present the integration and learning of must-link constraints in Section 6.3 and validate them experimentally under various setups in Section 6.4 on two video segmentation datasets (Brox and Malik, 2010; Galasso *et al.*, 2013).

6.2 PREVIOUS WORK ON MUST-LINK CONSTRAINTS

The usage of must-link constraints, first introduced by Wagstaff *et al.* (2001), is an active area of research in machine learning known as *constrained clustering* (see Basu *et al.* (2008) for an overview). The goal of integrating must-link constraints into spectral clustering has been tried via: **i**. modifying the value of affinities (cf. Kamvar *et al.* (2003), which first considered constrained spectral clustering); **ii**. modifying the spectral embedding (Li *et al.*, 2009); or **iii**. adding constraints in a post-processing step (Yu and Shi, 2001; Eriksson *et al.*, 2007; Xu *et al.*, 2009; Wang and Davidson, 2010; Maji *et al.*, 2011). Interestingly, none of these methods can guarantee that the must-link constraints are actually satisfied in the final clustering. By contrast, we employ must-link constraints to reduce the original graph to one of smaller size, thus enforcing the constraints while additionally benefiting runtime and memory consumption.

In particular, Rangapuram and Hein (2012) and Galasso *et al.* (2014) have shown that must-link constraints can be used to reduce the graph, based on the corresponding point groupings, and proved equivalence between the reduced and the original graph, respectively in terms of NCut (Rangapuram and Hein, 2012) and SC (Galasso *et al.*, 2014), for any clustering satisfying the must-link constraints. We employ these recent advances and propose to learn the must-link constraints in a data-driven discriminative fashion for video segmentation.

Other related work in segmentation have looked at merging superpixels with equivalence (Alpert *et al.*, 2012), but using hand-designed affinities, or learned pairwise relations between superpixels (Jain *et al.*, 2011), disregarding equivalence in the agglomerative merging process. This work brings together learning affinities and merging with equivalence guarantees for the first time.

6.3 LEARNING SPECTRAL MUST-LINK CONSTRAINTS

We provide here the steps of a video segmentation framework based on the normalized cut (Shi and Malik, 2000; Ng *et al.*, 2001; Hein and Setzer, 2011) and review the integration of must-link constraints by graph reductions as proposed in Rangapuram and Hein (2012); Galasso *et al.* (2014). While the idea of learning must-link constraints applies to any segmentation problem, we discuss in detail learning and inference in the specific case of the video segmentation features of Galasso *et al.* (2012).

6.3.1 Segmentation and Must-link Constraints

We represent a video sequence as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: nodes $i \in \mathcal{V}$ represent superpixels, extracted at each frame of the video sequence with an image segmentation algorithm (Arbeláez *et al.*, 2011); edges $e_{ij} \in \mathcal{E}$ between superpixels i and j take non-negative weights w_{ij} and express the similarity (*affinity*) between the superpixels.

A video segmentation can be defined as a partition $S = \{S_1, S_2, \ldots, S_K\}$ of the

(superpixel) vertex set $\mathcal{V}$, i.e. $\cup_k S_k = \mathcal{V}$, $S_k \cap S_m = \varnothing \ \forall \ k \neq m$. Given $\mathcal{S}$ the set of all partitions, we look for an optimal video segmentation $S^* = \{S_1^*, S_2^*, \ldots, S_N^*\} \in \mathcal{S}$ (where N is the number of visual objects), minimizer of an objective function, implicit (Grundmann *et al.*, 2010; Xu *et al.*, 2012; Paris, 2008) or explicit (Shi and Malik, 2000; Ng *et al.*, 2001; Vazquez-Reina *et al.*, 2010; Chang *et al.*, 2013).

Must-link constraints alter the video segmentation by reducing the set of feasible partitions $\mathcal{S}$. Given *correct*[2] must-links, a video segmentation algorithm generally improves in performance, since the solver is constrained to disregard non-optimal segmentations *wrt* S^*. Moreover, the integration of must-links leads to reduced runtime and memory load as the recent work (Rangapuram and Hein, 2012; Galasso *et al.*, 2014) suggests.

We are interested in learning a *must-link grouping function* $\mathcal{M}$, which groups *certain*[3] superpixels in the graph, while respecting S^*. $\mathcal{M}$ should *conservatively* associate each node i with a point grouping $I_k \subseteq S_l^*$ (in most uncertain cases a point grouping may only include a single node). More formally:

$$\begin{aligned} &\mathcal{M} : \mathcal{V} \mapsto \mathcal{P}, \quad i \mapsto I_k \\ &\qquad s.t. \ \ I_k \subseteq S_l^* \subseteq \mathcal{V}\,, \ \ \cup_k I_k = \mathcal{V}\,, \ \ I_k \cap I_m = \varnothing \ \ \forall \ \ k \neq m\,, \end{aligned} \tag{6.1}$$

where $\mathcal{P}$ is the set of possible partitions of $\mathcal{V}$.

6.3.2 Framework

Here we tailor the general theory to a video segmentation framework based on normalized cut, solved either via the spectral (Shi and Malik, 2000; Ng *et al.*, 2001) or 1-spectral (Bühler and Hein, 2009; Hein and Bühler, 2010) relaxation. Further, we discuss the integration of learned must-link constraints via graph reduction techniques (Rangapuram and Hein, 2012; Galasso *et al.*, 2014) and learning and inference strategies.

6.3.2.1 *Video segmentation setup*

We build upon Galasso *et al.* (2012). Their constructed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ uses superpixels extracted from the lowest level (level 1) of a hierarchical image segmentation (Arbeláez *et al.*, 2011). Edges connect superpixels from spatial and temporal neighbors and are weighted by their pair-wise affinities, computed from motion, appearance and shape features.

We consider six pairwise affinities: spatio-temporal appearance (STA), based on the median CIE Lab color distance; spatio-temporal motion (STM), based on median optical flow distance; across boundary appearance (ABA) and motion (ABM), computed across the common boundary of superpixels; short-term-temporal (STT),

[2] correct refers to the desired ground truth segmentation, which ideally corresponds with the optimal segmentation S^*

[3] certain groupings are the conservative grouping decisions which we propose to learn

measuring shape similarity by the spatial overlap of optical flow-propagated superpixels; long-term-temporal (LTT), given by the fraction of common trajectories between the superpixels. Additionally, we consider the number of common intersecting trajectories (IT). We distinguish four types of affinities, depending on whether the related superpixels: **i**. lie within the same frame (STA,STM,ABA,ABM); **ii**. lie on adjacent frames (STA,STM,STT); **iii-iv**. lie on frames at a distance of 2 (STT,LTT,IT) or more frames (LTT,IT) respectively.

6.3.2.2 *Video segmentation objective function*

Given a partition of $\mathcal{V}$ into N sets $S_1, \ldots, S_N$, the normalized cut (NCut) is defined (von Luxburg, 2007) as:

$$\mathrm{NCut}(S_1, \ldots, S_N) = \sum_{k=1}^{N} \frac{\mathrm{cut}(S_k, \mathcal{V} \backslash S_k)}{\mathrm{vol}(S_k)}, \tag{6.2}$$

where $\mathrm{cut}(S_k, \mathcal{V} \backslash S_k) = \sum_{i \in S_k, j \in \mathcal{V} \backslash S_k} w_{ij}$ and $\mathrm{vol}(S_k) = \sum_{i \in S_k, j \in \mathcal{V}} w_{ij}$. The balancing factor prevents trivial solutions and is ideal when unary terms cannot be defined, but is also the reason why minimization of the NCut is NP-Hard.

6.3.2.3 *Spectral relaxations*

The most widely adopted relaxation of NCut is spectral clustering (SC) (Shi and Malik, 2000; Ng *et al.*, 2001; von Luxburg, 2007), where the solution of the relaxed problem is given by representing the data points with the first few eigenvectors and then clustering them with k-means.

While widely adopted (Galasso *et al.*, 2014; Maire and Yu, 2013; Arbeláez *et al.*, 2011; Brox and Malik, 2010; Sundaram and Keutzer, 2011; Galasso *et al.*, 2012; Sundberg *et al.*, 2011), the SC relaxation is known to be *loose*. We therefore additionally consider the 1-spectral clustering (1-SC) (Hein and Bühler, 2010; Hein and Setzer, 2011) - a tight relaxation based on the 1-Laplacian. However, the relaxation is only tight for bi-partitioning, for multi-way partitioning recursive splitting is used as greedy heuristic. Reducing the original graph size with learned must-link constraints allows to experiment with 1-SC on two video segmentation benchmarks (Brox and Malik, 2010; Galasso *et al.*, 2013), notwithstanding the increased computational costs.

6.3.2.4 *Graph reduction schemes*

Given must-link constraints provided as point groupings $\{I_1, I_2, \ldots, I_q\}$ on the original vertex set $I_k \subseteq \mathcal{V}$, recent work (Rangapuram and Hein, 2012; Galasso *et al.*, 2014) shows how to integrate such constraints into the original problem with respectively preserving the NCut and the spectral clustering objective function.

In more detail, integration proceeds by reducing the original graph $\mathcal{G}$ to one of smaller size $\mathcal{G}^M = (\mathcal{V}^M, \mathcal{E}^M)$, whereby the vertex set is given by the point grouping $\mathcal{V}^M = \{I_1, I_2, \ldots, I_q\}$, the edge set $\mathcal{E}^M$ preserves the original node connectivity

and weights w_{IJ}^M are estimated so as to preserve the original video segmentation problem in terms of the NCut or spectral clustering objective. In particular, the NCut reduction is given by

$$w_{IJ}^M = \sum_{i \in I} \sum_{j \in J} w_{ij} \tag{6.3}$$

while the spectral clustering reduction is defined as

$$w_{IJ}^M = \begin{cases} \sum_{i \in I} \sum_{j \in J} w_{ij} & \text{if } I \neq J \\ \frac{1}{|I|} \sum_{i \in I} \sum_{j \in J} w_{ij} - \frac{(|I|-1)}{|I|} \sum_{i \in I} \sum_{j \in \mathcal{V} \setminus I} w_{ij} & \text{if } I = J, \end{cases} \tag{6.4}$$

provided equal affinities of elements of $\mathcal{G}$ constrained in $\mathcal{G}^M$, cf. Galasso *et al.* (2014).

6.3.3 Learning

An ideal must-link constraining function $\mathcal{M}$ (Eq. 6.1) should only merge superpixels which are *correct*, i.e. belong to the same set in the optimal segmentation. From an implementation viewpoint, it is convenient to consider instead $\mathcal{M}_{pw}$, defined over the set of edges $\mathcal{E}$ of the graph $\mathcal{G}$ representing the video sequence:

$$\mathcal{M}_{pw} : \mathcal{E} \mapsto \{0, 1\} \tag{6.5}$$

$\mathcal{M}_{pw}$ casts the must-link constraining problem as a binary classification one, where a TRUE output for an input edge e_{ij} means that i and j belong to the same point grouping, in the must-link constrained graph $\mathcal{G}^M$.

We learn $\mathcal{M}_{pw}$ with Random Forests (Breiman, 2001; Criminisi *et al.*, 2012) using as features the affinities of Galasso *et al.* (2012) (STA,STM,ABA,ABM,STT,LTT) and the additional IT which we described in Section 6.3.2.1. Since different sets of affinities are available depending on whether two superpixels lie on the same or on different frames, we learn 4 different classifiers to match the 4 types of affinities.

We train a set of independent trees by estimating optimal parameters θ_p for the split functions $h(x, \theta_p)$ at each tree node p, as a function of the computed features x. Given a training set $T_p \subset X \times Y$, with X the vector of computed features and $Y = \{0, 1\}$ the corresponding ground truth video annotations, we seek to maximize the information gain I_p:

$$I_p(T_p, T_p^L, T_p^R) = H(T_p) - \frac{|T_p^L|}{|T_p|} H(T_p^L) - \frac{|T_p^R|}{|T_p|} H(T_p^R), \tag{6.6}$$

with $T_p^L = \{(x, y) \in T_p | h(x, \theta_p) = 0\}$, $T_p^R = T_p \setminus T_p^L$, the Shannon entropy $H(T) = -\sum_{y \in \{0,1\}} p_y \log(p_y)$ and p_y is the pdf of outcome y.

We extend the formulation of (6.6) to allow for learning must-link constraints on pre-grouped nodes. Galasso *et al.* (2014) use superpixel groupings (larger superpixel named *level 2*, cf. 6.4). It is important, as we found out, to consider the node

multiplicity. We define therefore $|T_p| = \sum_{k \in T_p} m_k$, where $m_k = |I_k| \cdot |J_k|$ is the multiplicity of the edge between superpixel groupings I_k and J_k, thus $p_y = \frac{\sum_y m_y}{\sum_{y \in \{0,1\}} m_y}$.

Must-link constraints have a transitive nature: $\mathcal{M}_{pw}(e_{ij}) = 1$ and $\mathcal{M}_{pw}(e_{ik}) = 1$ imply $\mathcal{M}_{pw}(e_{jk}) = 1$. It is therefore crucial that all decided constraints are correct, as a few wrong ones may result in a larger set of incorrect decisions by transitive closure and potentially spoil the segmentation. Thus we define the hyper-parameters (threshold of the classifier and tree depth) such that $\mathcal{M}_{pw}$ provides the largest number of positive predictions (the must-link decisions), while making zero false positives on the validation set. In such a conservative way we ensure that the resulting classifier makes only a very small number of false positives on unseen data. Although this conservative classifier might imply that in the worst case, no must-link constraints are predicted, it turns out our classifier actually predicts for a large fraction of the edges to be linked and thus leads to a significant reduction in size, while making a few false positives on the unseen data (overall, 1 false positive per 242k true predictions).

6.3.4 Inference

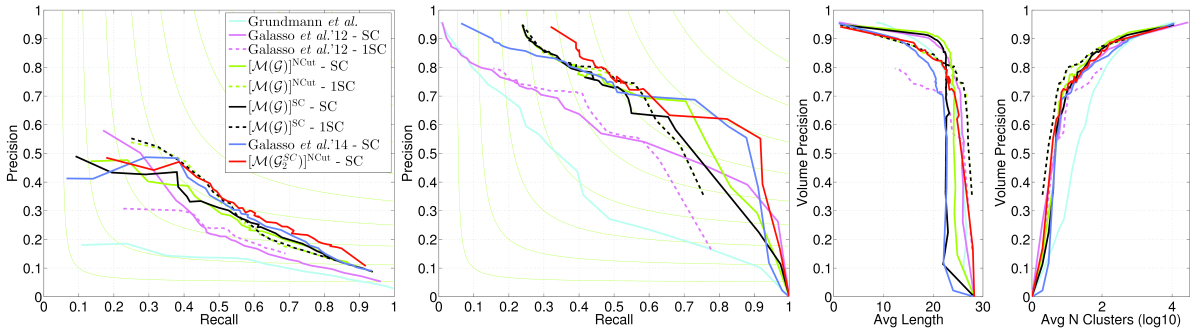
The learned must-link constraining function $\mathcal{M}_{pw}$ provides must-link decisions for each edge of graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. A further propagation of merge decisions in the graph accounts for the transitivity closure of $\mathcal{M}_{pw}$, consistently with the validation procedure (cf. Section 6.3.3). Based on the must-link decisions, we use the graph reduction techniques of Section 6.3.2.4, which integrate must-link decisions into graph $\mathcal{G}$ by reducing it to the smaller one $\mathcal{G}^M = (\mathcal{V}^M, \mathcal{E}^M)$ based on the determined groupings.

The described framework allows for evaluating different reduction schemes (equivalence in terms of NCut (Rangapuram and Hein, 2012) and SC (Galasso *et al.*, 2014)) and various spectral partitioning functions (1-SC (Hein and Setzer, 2011) and SC (Shi and Malik, 2000; Ng *et al.*, 2001)). It further allows to include spatial must-link constraints and use larger superpixels, as done in Galasso *et al.* (2014). We report experimental results on all these combinations in the following section.

6.4 EXPERIMENTAL EVALUATION

We conduct two sets of experiments to analyze performance and efficiency of must-link constrained graphs $\mathcal{G}^M$. In both cases we adopt the recently proposed benchmark metrics of Galasso *et al.* (2013): the boundary precision-recall (BPR) from Arbeláez *et al.* (2011) and the volume precision-recall (VPR) metric. Besides the PR curves, we report aggregate performance for BPR and VPR: optimal dataset scale [ODS], optimal segmentation scale [OSS], average precision [AP].

In the first set of experiments, we consider the *Berkeley Motion Segmentation Dataset* (BMDS) (Brox and Malik, 2010), which consists of 26 VGA-quality video sequences,



BPR VPR Length NCL

	BPR			VPR			Length	NCL
Algorithm	ODS	OSS	AP	ODS	OSS	AP	$\mu(\delta)$	μ
Grundmann *et al.* (Grundmann *et al.*, 2010)	0.22	0.25	0.12	0.42	0.44	0.39	26.06(6.34)	13.81
Galasso *et al.* (Galasso *et al.*, 2012) - SC	0.37	0.39	0.24	0.57	0.72	0.59	25.75(6.46)	4.00
$[\mathcal{M}(\mathcal{G})]^{\text{NCut}}$ - SC	0.40	0.45	0.26	**0.69**	**0.77**	**0.69**	24.17(8.57)	6.00
$[\mathcal{M}(\mathcal{G})]^{\text{SC}}$ - SC	**0.41**	**0.46**	**0.27**	0.64	0.75	0.67	22.66(9.55)	6.00
Galasso *et al.* (Galasso *et al.*, 2012) - 1SC	0.34	0.36	0.19	0.56	0.62	0.49	25.99(6.61)	5.00
$[\mathcal{M}(\mathcal{G})]^{\text{NCut}}$ - 1SC	**0.44**	**0.48**	**0.34**	**0.64**	0.70	**0.60**	26.62(5.80)	5.00
$[\mathcal{M}(\mathcal{G})]^{\text{SC}}$ - 1SC	0.43	**0.48**	**0.34**	**0.64**	**0.71**	**0.60**	26.41(5.95)	5.00
Galasso *et al.*'14 (Galasso *et al.*, 2014) - SC	**0.43**	**0.48**	**0.29**	**0.71**	0.79	0.71	22.04(8.92)	7.00
$[\mathcal{M}(\mathcal{G}_2^{SC})]^{\text{NCut}}$ - SC	**0.43**	**0.48**	0.28	**0.71**	**0.80**	**0.75**	24.77(7.49)	5.00

Figure 6.2: Comparison of video segmentation algorithms with the learned must-links, on BMDS (restricted to first 30 frames) (Brox and Malik, 2010). The plots and table show BPR and VPR, aggregate measures ODS, OSS and AP, and length statistics (mean μ, std. δ, no. clusters NCL) (Galasso *et al.*, 2013).

representing mainly humans and cars, which we arrange into training, validation and test sets (6+4+16). We restrict sequences to the first 30 frames. The ground truth is provided for the 1st, 10th, 20th, 30th frame. We further annotate the 2nd, 9th, 11th frame to learn must-links across 1 and 2 frames (the extra annotations are public now).

We compare the baseline of Galasso *et al.* (2012) with the proposed variants, $[\mathcal{M}(\mathcal{G})]^{\text{NCut}}$ - SC and $[\mathcal{M}(\mathcal{G})]^{\text{SC}}$ - SC, reducing the original graph $\mathcal{G}$ of Galasso *et al.* (2012) with learned must-links to $\mathcal{G}^M$ by using respectively the normalized cut (NCut) and spectral clustering (SC) reductions, and then performing SC. Figure 6.2 (*plots*) shows that both proposed variants outperform the baseline algorithm (Galasso *et al.*, 2012) both on BPR and VPR. The table shows improvement by 4.7% in BPR and 9% in VPR. Since the average number of superpixels is reduced by 66.7%, the better performance is accompanied by a reduction of 60% in runtime and 90% in memory load.

In Figure 6.2, we further experiment by adopting 1-spectral clustering (1-SC) (Hein and Setzer, 2011) for the NCut within the baseline algorithm (Galasso *et al.* (2012)

- 1-SC), and we compare this with our proposed variants, $[\mathcal{M}(\mathcal{G})]^{\text{NCut}}$ - 1-SC and $[\mathcal{M}(\mathcal{G})]^{\text{SC}}$ - 1-SC, where we have grouped superpixels according to learned must-links prior to processing (here with 1-SC). Since 1-SC is more costly, the provided computational reduction is even more desirable here. Again, our proposed variants improve in performance, as it appears both in the plots and the tables (average improvement of 12.3% in BPR and 9% in VPR), while significantly reducing runtime (improved by 80%) and memory load (improved by 90%). We note the similar performance of 1-SC for both reduction variants, $[\mathcal{M}(\mathcal{G})]^{\text{NCut}}$ and $[\mathcal{M}(\mathcal{G})]^{\text{SC}}$, which surprises because only the NCut reduction is theoretically justified in combination with 1-SC. Moreover, we observe the better performance of SC over 1-SC. This may indicate that the affinities of Galasso *et al.* (2012), designed for SC, do not fit as well the original (but different) NCut problem.

Additionally, we consider the recent work of Galasso *et al.* (2014), which uses superpixels extracted from a higher hierarchical level of an image segmentation algorithm (Arbeláez *et al.*, 2011) (superpixels at level 2), computes affinities between them and re-weights them according to SC, to take the finest superpixels at level 1 into account. Our proposed method based on must-links also allows learning constraints on the larger superpixel graph $\mathcal{G}_2$ (the multiplicity of point groupings plays a role in this case, cf. Section 6.3.3). Figure 6.2 shows that the reduction $[\mathcal{M}(\mathcal{G}_2^{SC})]^{\text{NCut}}$ - SC leads to the same performance as the original algorithm (Galasso *et al.*, 2014) on BPR and improves on VPR, while reducing the problem size *wrt* Galasso *et al.* (2014) (runtime by 30% and memory load by 70%).

Figure 6.3 qualitatively supports the positive results. Note that the learned must-links respect the GT objects while reducing the number of employed superpixels, $\mathcal{M}$ SPX. Improvements in the video segmentation output ($\mathcal{M}$ Segm Vs. (SPX) Segm.) are more evident for 1-SC. The proposed learned must-links determine merging both in the spatial and temporal dimension. It is interesting to note that for the BMDS (Brox and Malik, 2010) most merging comes from the first: it seems easier to make conservative merging assumptions within the frame.

In the second set of experiments we consider the benchmark VSB100 (Galasso *et al.*, 2013), which includes 100 HD quality videos (Sundberg *et al.*, 2011) arranged into train and test sets (40+60) (we split training – 24 – and validation set – 16). In Figure 6.4 we compare the proposed method $[\mathcal{M}(\mathcal{G}_2^{SC})]^{\text{NCut}}$ - SC to the baseline (Galasso *et al.*, 2014) and other video segmentation algorithms. Our method maintains the performance of Galasso *et al.* (2014) on BPR and slightly improves on VPR. This shows that Galasso *et al.* (2014), by jointly leveraging large powerful superpixels (Arbeláez *et al.*, 2011), *saturate* the few affinities of Galasso *et al.* (2012), which we also use here. Thus learned must-links closely follow the spectral clustering optimization and our proposed method only provides further reduction of the problem size. With similar arguments, as also maintained in Galasso *et al.* (2014), the segmentation propagation method of Galasso *et al.* (2013) is only partially outperformed, due to its more complex image features e.g. textures. Both observations suggest to use more complex features for learning. With respect to the efficient reduction of Galasso *et al.* (2014), we further reduce runtime by 30% and memory load by 65%, while we

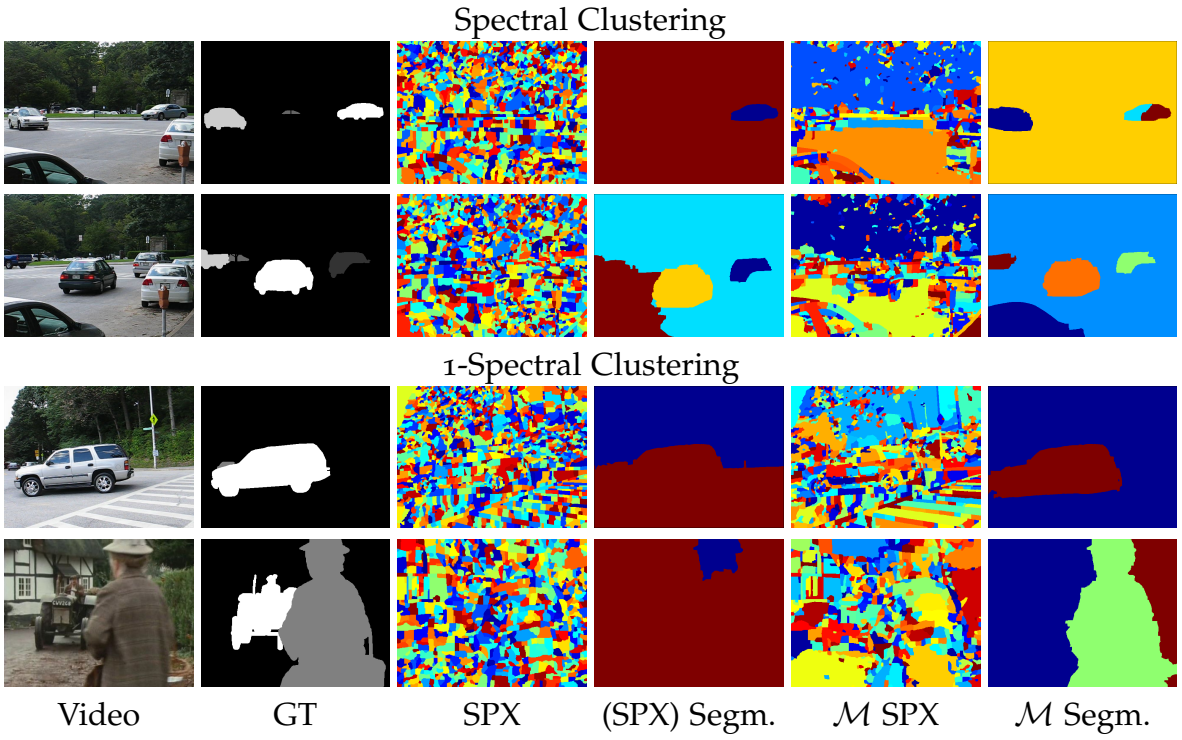
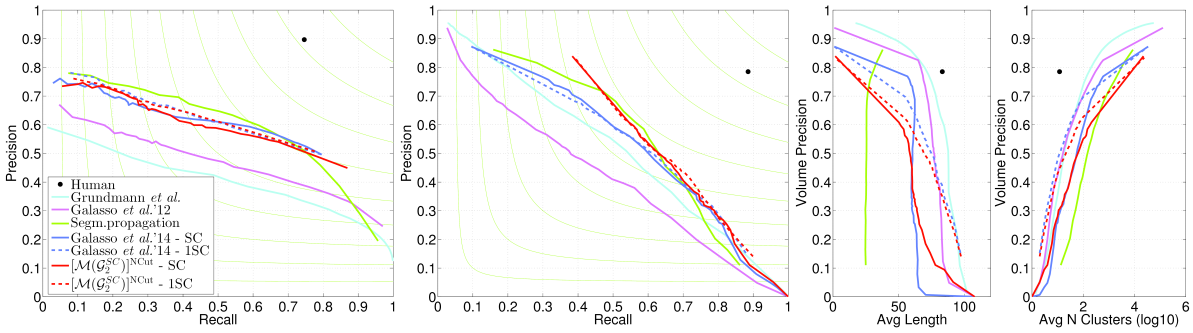


Figure 6.3: Sample superpixels (SPX) and segmentation results of Galasso *et al.* (2012), compared with the proposed learned must-link variants, both when employing SC and 1-SC (cf. Section 6.4 for details). The proposed superpixels ($\mathcal{M}$ SPX) respect the video segmentation output while reducing the problem size. Additionally, $\mathcal{M}$ SPX improve results, *esp.* for 1-SC.

reduce runtime by 97% and memory load by 87% *wrt* Galasso *et al.* (2012).

In addition, we adopt 1-spectral clustering (Hein and Setzer, 2011) within the baseline (Galasso *et al.* (2014) - 1-SC), and compare this with our proposed method ($[\mathcal{M}(\mathcal{G}_2)]^{\text{NCut}}$ - 1SC). Figure 6.4 shows that $[\mathcal{M}(\mathcal{G}_2^{SC})]^{\text{NCut}}$ - 1SC results in the same performance on BPR and minor improvement on VPR, while significantly reducing runtime (by 70%) and memory load (by 65%) *wrt* Galasso *et al.* (2014).

IMPLEMENTATION DETAILS. We use the Random Forests implementation of Criminisi *et al.* (2012). The number of features to sample for each node split is set to $\sqrt{\text{F}}$, where F is the dimensionality of the feature space. The averaged prediction of the individual trees is taken for prediction of the ensemble. As weak learners we use linear binary split functions and conic sections, and the forest size is set to 100 trees. The tree depth is varied in the range $[2, 12]$ and validated along with the threshold, which yields the largest number of must-links with zero false positives. Following Galasso *et al.* (2012), we extract the first 6 eigenvectors.



BPR VPR Length NCL

	BPR			VPR			Length	NCL
Algorithm	ODS	OSS	AP	ODS	OSS	AP	$\mu(\delta)$	μ
Human	0.81	0.81	0.67	0.83	0.83	0.70	83.24(40.04)	11.90
Grundmann *et al.* (Grundmann *et al.*, 2010)	0.47	0.54	0.41	0.52	0.55	0.52	87.69(34.02)	18.83
Galasso *et al.*'12 (Galasso *et al.*, 2012)	0.51	0.56	0.45	0.45	0.51	0.42	80.17(37.56)	8.00
Segm. propagation (Galasso *et al.*, 2013)	0.61	0.65	**0.59**	**0.59**	**0.62**	0.56	25.50(36.48)	258.05
Galasso *et al.*'14 (Galasso *et al.*, 2014) - SC	**0.62**	0.65	0.50	0.55	0.59	0.55	61.25(40.87)	80.00
$[\mathcal{M}(\mathcal{G}_2^{SC})]^{NCut}$ - SC	0.61	**0.66**	**0.52**	**0.58**	**0.61**	**0.58**	51.72(39.90)	176.65
Galasso *et al.*'14 (Galasso *et al.*, 2014) - 1SC	**0.61**	**0.64**	**0.52**	0.55	0.60	0.54	69.80(42.26)	19.00
$[\mathcal{M}(\mathcal{G}_2^{SC})]^{NCut}$ - 1SC	**0.61**	**0.64**	0.51	**0.58**	**0.61**	**0.58**	60.48(43.19)	50.00

Figure 6.4: Comparison of video segmentation algorithms with our proposed method based on the learned must-links, on VSB100 (Galasso *et al.*, 2013) (cf. Section 6.4 for details).

6.5 CONCLUSIONS

We have formalized must-link constraints and proposed the relevant learning and inference algorithms. While this theory is applicable to general clustering and segmentation problems, we have particularly shown the use of learned must-link constraints in conjunction with spectral techniques, whereby recent theoretical advances employ these to reduce the original problem size, hence the runtime and memory requirements. Experimentally, we have shown that learned must-link constraints improve efficiency and, in most cases, performance, as these allow discriminatively training on GT data.

In Chapter 7 we show how to construct a graph in order to improve video segmentation performance as well as reduce the problem size without changing the graph partitioning model.

7 CLASSIFIER BASED GRAPH CONSTRUCTION FOR VIDEO SEGMENTATION

While a wide variety of features has been explored and various graph partition algorithms have been proposed, there is surprisingly little research on how to construct a graph to obtain the best video segmentation performance. This is the focus of this chapter.

We propose to combine features by means of a classifier, use calibrated classifier outputs as edge weights and define the graph topology by edge selection. By learning the graph (without changes to the graph partitioning method), we improve the results of the best performing video segmentation algorithm by 6% on the challenging VSB100 benchmark, while reducing its runtime by 55%, as the learnt graph is much sparser.

7.1 INTRODUCTION

Video segmentation has recently witnessed growing interest (Banica *et al.*, 2013; Chang *et al.*, 2013; Fragkiadaki and Shi, 2012; Jain *et al.*, 2013; Li *et al.*, 2013; Maire and Yu, 2013; Palou and Salembier, 2013; Reso *et al.*, 2013; Zhang *et al.*, 2013). On the one hand, this is motivated by its usefulness for applications such as semantic scene understanding (Jain *et al.*, 2013), activity recognition (Taralova *et al.*, 2014), or geometric context classification (Raza *et al.*, 2013). In these cases, organizing a video into spatio-temporal tubes allows the joint consideration of appearance and motion, while reducing the search space for the solution. On the other hand, video segmentation poses interesting research questions. In addition to the scene and scale ambiguities of image segmentation (Arbeláez *et al.*, 2011; Dollár and Zitnick, 2015; Isola *et al.*, 2014; Ren and Bo, 2012), various parts of the scene will change over time as well as appear or disappear.

Graph-based approaches are among the top-performing methods for video segmentation (Grundmann *et al.*, 2010; Fragkiadaki and Shi, 2012; Palou and Salembier, 2013; Xu *et al.*, 2013; Galasso *et al.*, 2014). The use of graphs is long established in segmentation (Shi and Malik, 2000; Arbeláez *et al.*, 2011; Isola *et al.*, 2014; Maire and Yu, 2013; Brox and Malik, 2010; Palou and Salembier, 2013; Sundaram and Keutzer, 2011). Graphs provide a natural representation of image/video sequences, where edges encode the spatio-temporal structure, and allow long-term reasoning due to their transitivity property. Graph-based video segmentation techniques:

1. compute features among pairs of pixels or superpixels;

2. design a graph according to the spatio-temporal neighborhood of the pixels or

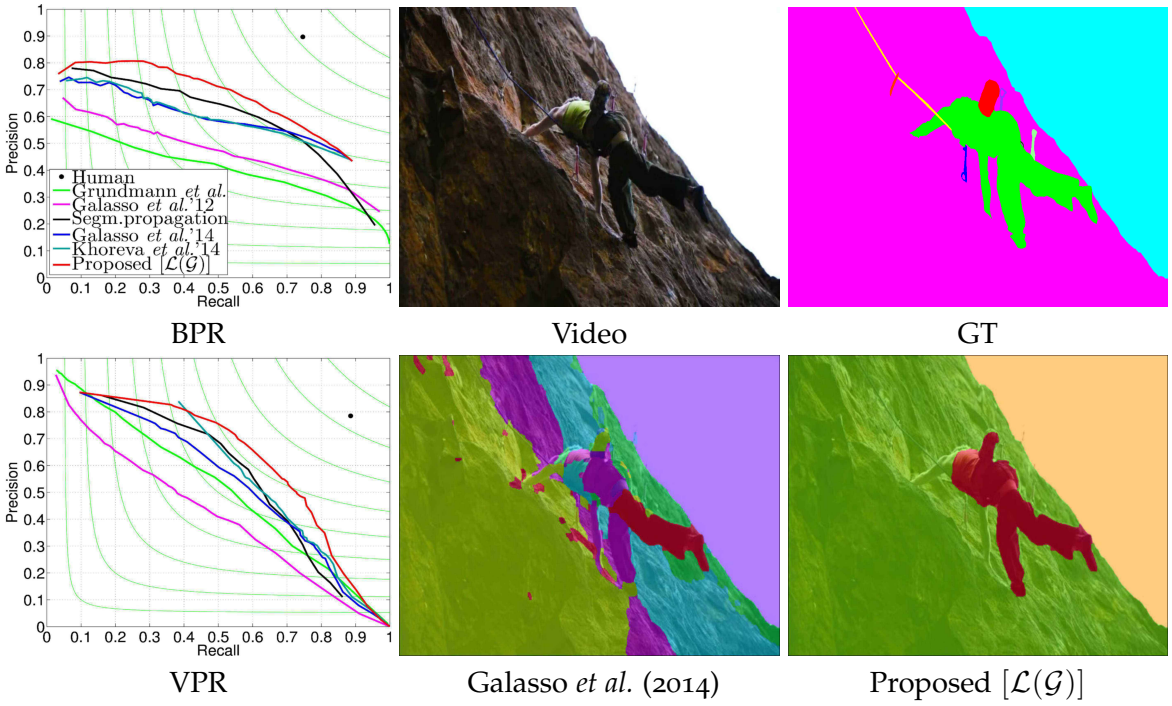


Figure 7.1: Climbing up! We contribute theory and best-practices for *graph construction* and achieve high-quality results on the challenging VSB100 Galasso *et al.* (2013) (BPR and VPR reported here, more details in Section 7.5.)

superpixels and manually combine features to weight its edges;

3. partition the graphs with spatio-temporal clustering.

Previous work has used a variety of features and has proposed various graph partitioning algorithms. However, we argue in this chapter that constructing the underlying graph is a crucial step for best performance of such graph-based methods that has received little attention in the literature. This work therefore explicitly addresses the problem of graph construction. We propose and empirically evaluate procedures and validation-based best practices to learn both the edge topology and weights.

Our contribution includes:

- Using a classifier for learning the pairwise similarities between superpixels, leveraging the recent availability of a larger training set for video segmentation (Galasso *et al.*, 2013).
- Employing different classifiers for differently-neighboring superpixels (within the same frame or across time) and further considering the neighboring topology (superpixels directly neighboring or connected by longer-term links).
- Calibrating the confidence of the various classifiers with their classification accuracy.

- Selecting edges based on the classifier confidence which, while further improving the quality, also reduces the graph size and thus the computational load.

These topics are respectively treated in Section 7.4. In Section 7.3 we present the features and the graph partitioning model we use. The proposed approach based on learning allows the seamless integration of multiple features from recent literature (Brox and Malik, 2010; Arbeláez *et al.*, 2011; Palou and Salembier, 2013; Galasso *et al.*, 2014). We build upon the graph partitioning model of Galasso *et al.* (2014) based on spectral clustering and show that addressing the graph construction *explicitly* helps to achieve better performance (cf. Figure 7.1) without altering the graph partitioning or the underlying features.

7.2 PREVIOUS WORK ON GRAPH CONSTRUCTION

Meaningful features are necessary for good video segmentation. Much literature (Brox and Malik, 2010; Grundmann *et al.*, 2010; Palou and Salembier, 2013; Galasso *et al.*, 2012) has proposed features for appearance, motion or shape similarities among the graph nodes. Most works are currently limited in the number of features they can leverage, as often the researchers hand-design the feature combination to measure similarity between pixels or superpixels. In this work we learn classifiers to combine features and seamlessly integrate them.

Much research has been devoted to graph partitioning models (Couprie *et al.*, 2011; Xu *et al.*, 2013; Maire and Yu, 2013; Andres *et al.*, 2011; Cheng and Ahuja, 2012; Jain *et al.*, 2013; Galasso *et al.*, 2014). While measurable differences have been observed we intentionally focus on the graph construction problem instead. Therefore, we adopt the recent and successful graph partitioning model (Galasso *et al.*, 2014), which is based on spectral clustering (Ng *et al.*, 2001; Shi and Malik, 2000; Brox and Malik, 2010; Fragkiadaki and Shi, 2012; Sundaram and Keutzer, 2011). However, our proposed graph construction is directly applicable to other graph-based techniques (see Section 7.5).

Constructing the graph is a vital step for ensuring the performance of clustering methods (Maier *et al.*, 2009; Jebara and Chang, 2009) Although graph-based methods have been extensively studied, there have been limited efforts for building effective graphs. The most popular method for constructing a sparse graph is the nearest neighbor (NN) approach, including different variants such as k-nearest neighbor and ϵ-nearest neighbor methods. Another contender approach is the b-matching procedure (Jebara and Shchogolev, 2006), which prunes graph edges such that the degree of each node is b, producing a more balanced variant of k-nearest neighbor. Several works explored semi-supervised learning of the graph (Jebara and Chang, 2009; Alexandrescu and Kirchhoff, 2007), i.e. learning the graph from its partial labelling. By contrast our method is applied to unlabeled test-set videos.

To the best of our knowledge, graph construction based on classifier-learnt combination of features is novel in video segmentation. While learning the edge

weights of the graph has been exploited in image segmentation (Ren and Malik, 2003; Turaga *et al.*, 2009; Kim *et al.*, 2013), our work addresses the topology of the graph, raising novel issues, such as weight-calibration and edge-selection, which we discuss in Section 7.4. Learning the topology provides larger performance gains and benefits efficiency due to a sparser structure of the constructed graph.

7.3 GRAPH-BASED VIDEO SEGMENTATION

Let us represent a video sequence as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Nodes $i \in \mathcal{V}$ are superpixels, extracted at each frame from a specific hierarchical level of an image segmentation algorithm (Arbeláez *et al.*, 2011). Following Galasso *et al.* (2014), we do not consider the lowest level (finest superpixels), but rather extract them by thresholding the ultrametric contour map (*ucm*) at a higher value (0.12), whereby fewer of them (*larger superpixels*) provide a comparable model error.

Edges of the graph $e_{ij} \in \mathcal{E}$ connect pairs of superpixels i and j with non-negative weight w_{ij}, which expresses their similarity. Following Grundmann *et al.* (2010); Lezama *et al.* (2011); Galasso *et al.* (2012); Palou and Salembier (2013), edges may connect neighbors:

within frame: i and j are neighbors if they share a common part of their superpixel contour or are close by in the spatial domain of the frame;

across 1 frame: connected by coordinate correspondences over time;

across 2 frames: connected by across-1 correspondences, further propagated over one more frame;

across $>$ 2 frames: linked if overlapping with common long-term point trajectories.

Graph based video segmentation proceeds in three main steps:

1. *Feature computation.* Depending on the edge type, a number of features are available to compute the similarity between superpixels. For example, superpixels on the same frame may be related by the strength of the image segmentation boundary between them (*aba*) and by the χ^2 distance between their color histograms (sta_{χ^2}); if neighboring across frames, just sta_{χ^2} applies (see Section 7.4.1 and Table 7.1 for more details).

2. *Graph construction.* State-of-the-art approaches use edges e_{ij} if the two superpixels are neighbors, either within or across frames. Then, they compute edge weights w_{ij} by combining the similarities from the applicable features linearly. Current video segmentation literature (Brox and Malik, 2010; Grundmann *et al.*, 2010; Galasso *et al.*, 2012; Palou and Salembier, 2013) sets the combinations manually on a validation set.

3. *Graph partitioning.* Video segmentation S is defined as a partition of the vertex set $\mathcal{V}$: $S = \{S_1, S_2, \ldots, S_K\}$, i.e. $\cup_k S_k = \mathcal{V}$, $S_k \cap S_m = \varnothing \;\; \forall \;\; k \neq m$. Given $\mathcal{S}$ the set of all partitions, graph partitioning looks for the optimal video segmentation $S^* = \{S_1^*, S_2^*, \ldots, S_N^*\} \in \mathcal{S}$ (where N is the number of visual objects) which minimizes an objective function, implicitly (Grundmann *et al.*, 2010; Xu *et al.*, 2012; Paris, 2008) or explicitly (Shi and Malik, 2000; Ng *et al.*, 2001; Vazquez-Reina *et al.*, 2010; Chang *et al.*, 2013).

Different to previous work, we focus on the *graph construction*. Since we use discriminatively trained classifiers to combine features, we name ours a *learnt graph* $\mathcal{L}(\mathcal{G})$. Furthermore, we investigate graph topology, classifier output confidence mapping and edge selection in detail in Section 7.4.

In the rest of this Section, we present the *features* which we use and the *graph partitioning* model which we adopt, based on spectral clustering and the graph-equivalent reweighting from Galasso *et al.* (2014). We use the publicly available code of Galasso *et al.* (2014) for the original graph construction and partitioning method.

7.3.1 Superpixel features

Adopting learning allows to seamlessly integrate an arbitrary number of features into the computation of the graph edge weights, *letting the classifier work out the optimal combination*. We consider 14 well-established features from video segmentation techniques (Hoiem *et al.*, 2007; Brox and Malik, 2010; Arbeláez *et al.*, 2011; Galasso *et al.*, 2012; Palou and Salembier, 2013; Galasso *et al.*, 2014), which apply to superpixels. We present them by grouping appearance, motion and shape features.

7.3.1.1 *Appearance Based Features*

Across boundary appearance [*aba*]. This measures similarity in the close vicinity of the common boundary between two superpixels i_f and j_f by averaging the common boundary strength (here and in the following we explicitly indicate the frame f which the superpixel belongs to for clarity). We take $\overline{v}_f^{ij}$ the average ultrametric contour map of Arbeláez *et al.* (2011) as a measure of the boundary strength between i and j and define: $aba(i_f, j_f) = \overline{v}_f^{ij}$.

Spatio-temporal appearance [sta, sta_{χ^2}]. This uses the distance between the median brightness and color of a superpixel in *Lab*-color-space as a measure of the overall similarity among two superpixels i and j, from the same or different frames f and f': $sta(i_f, j_{f'}) = \exp\left\{-\lambda_{sta} \| \overline{Lab}_{i_f} - \overline{Lab}_{j_{f'}} \|\right\}$.

Similarly sta_{χ^2} measures the overall appearance similarity using *Lab* (8-bin) color histograms and their χ^2 distance: $sta_{\chi^2}(i_f, j_{f'}) = \exp\left\{-\lambda_{sta_{\chi^2}} d_{\chi^2}(h(Lab_{i_f}), h(Lab_{j_f}))\right\}$.

Texture [$text$, $text_{\chi^2}$]. Texture information may be encoded (cf. Hoiem *et al.* (2007) for more details) with (a subset of) the textons designed by Leung and Malik (2001). We consider the L_2 distance between the mean absolute filter responses $text(i_f, j_{f'}) =$

$\exp\{-\lambda_{text}\|\overline{T}_{i_f} - \overline{T}_{j_{f'}}\|\}$ and the chi-squared distance between the histograms of maximum filter responses $text_{\chi^2}(i_f, j_{f'}) = \exp\{-\lambda_{text_{\chi^2}} d_{\chi^2}(h(T_{i_f}), h(T_{j_f}))\}$.
Size ratio [*size*]. We further consider the relative size difference of superpixels as an indication of appearance similarity $size(i_f, j_{f'}) = \big||i_f| - |j_{f'}|\big| / \max\{|i_f|, |j_{f'}|\}$.

7.3.1.2 *Motion Based Features*

Across boundary motion [*abm*]. We consider an optical flow estimate (Zach *et al.*, 2007), which we smooth spatially (preserving the across-superpixels boundaries with bilateral filtering) and temporally (median filtered ± 2 frames). The resulting $\overline{u}^f(x)$ (simply indicated as $\overline{u}^f$ in the following) allows to compute the motion similarity in the vicinity of the boundary between two superpixels by averaging their $\overline{u}^f$ distance across the common boundary ψ_f^{ij}:
$abm(i_f, j_f) = \exp\{-\lambda_{abm}\big(\sum_{(x_i^m, x_j^m)\in\psi_f^{ij}} \|\overline{u}^f(x_i^m) - \overline{u}^f(x_j^m)\|^2\big) / |\psi_f^{ij}|\}$.
Spatio-temporal motion [*stm*, stm_{χ^2}]. This measures the overall motion similarity between two superpixels i_f and $j_{f'}$ based on their median optical flow $\overline{u}$: $stm(i_f, j_{f'}) = \exp\{-\lambda_{stm}\|\overline{u}_{i_f} - \overline{u}_{j_{f'}}\|^2\}$.

Similarly, we may compute the similarity with the χ^2 distance between the superpixel optical flow (22 bin) histograms: $stm_{\chi^2}(i_f, j_{f'}) = \exp\{-\lambda_{stm_{\chi^2}} d_{\chi^2}\big(h(u_{i_f}), h(u_{j_{f'}})\big)\}$.
Spatial distance [*sd*]. As a measure of motion-displacement, we additionally consider the spatial distance between centroids of superpixels c_{i_f} and $c_{j_{f'}}$ across frames: $sd(i_f, j_{f'}) = \|c_{i_f} - c_{j_{f'}}\|$.

7.3.1.3 *Shape Based Features*

Short term temporal [*stt*]. We measure the shape similarity by comparing $m_{j_{f'}}$ the *shape* (its binary mask m) of a superpixel j at frame f' with the shape of i_f propagated with optical flow to frame f' (its projected mask $m_{i_f}^{f'}$). *stt* is given by the Dice coefficient between the true $m_{j_{f'}}$ and optical-flow-projected $m_{i_f}^{f'}$ binary mask:
$stt(i_f, j_{f'}) = 2|m_{i_f}^{f'} \bigcap m_{j_{f'}}| / \big(|m_{i_f}^{f'}| + |m_{j_{f'}}|\big)$.
Long term temporal [*ltt*, *cit*, *td*]. In a similar spirit to *stt*, *ltt* measures the similarity between superpixels i_f and $j_{f'}$ which belong to frames potentially further in time from each other ($f' = f + m$, $m \in (0, F]$ where F scales up to the whole length of the video sequence). We consider the dense point trajectories of (Sundaram *et al.*, 2010) as a measure of the shape (binary mask) projection. Let Φ_{i_f} be the subset of trajectories intersecting superpixel i_f. The similarity is the Dice measure between the intersection sets Φ_{i_f} and $\Phi_{j_{f'}}$: $ltt(i_f, j_{f'}) = 2\big|\Phi_{i_f} \bigcap \Phi_{j_{f'}}\big| / \big(\big|\Phi_{i_f}\big| + \big|\Phi_{j_{f'}}\big|\big)$.

We additionally provide the classifier with the number of common intersecting trajectories (the fewer dense tracks are available, the less it should rely on *ltt* as a reliable shape similarity): $cit(i_f, j_{f'}) = \big|\Phi_{i_f} \bigcap \Phi_{j_{f'}}\big|$ and temporal distance between superpixels $td(i_f, j_{f'}) = |f - f'|$.

7.3.2 Graph partitioning

We consider the graph partitioning model of Galasso *et al.* (2014). The approach seeks to determine the graph partition $S = \{S_1, S_2, \ldots, S_N\}$ (complete and disjoint $\cup_k S_k = \mathcal{V}$, $S_k \cap S_m = \varnothing \ \forall \ k \neq m$) which is optimal according to the *normalized cut* (NCut) objective:

$$\mathrm{NCut}(S_1, \ldots, S_N) = \sum_{k=1}^{N} \frac{\mathrm{cut}(S_k, \mathcal{V} \backslash S_k)}{\mathrm{vol}(S_k)}, \tag{7.1}$$

where $\mathrm{cut}(S_k, \mathcal{V} \backslash S_k) = \sum_{i \in S_k, j \in \mathcal{V} \backslash S_k} w_{ij}$ and $\mathrm{vol}(S_k) = \sum_{i \in S_k, j \in \mathcal{V}} w_{ij}$.

Following established literature (Shi and Malik, 2000; Ng *et al.*, 2001; von Luxburg, 2007; Corso *et al.*, 2008; Brox and Malik, 2010; Arbeláez *et al.*, 2011; Sundberg *et al.*, 2011; Sundaram and Keutzer, 2011; Di *et al.*, 2012; Galasso *et al.*, 2012; Fragkiadaki and Shi, 2012; Maire and Yu, 2013; Galasso *et al.*, 2014), we consider the spectral relaxation of the NCut problem (otherwise NP-Hard):

$$\min_{T} \mathrm{Tr}(T' L_{\mathrm{sym}} T) \quad \text{subject to} \quad TT' = I, \, T = D^{\frac{1}{2}} H, \tag{7.2}$$

where H is the matrix containing indicator vectors h_i, $L_{\mathrm{sym}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ is the normalized graph Laplacian, W is the matrix containing the pairwise affinities w_{ij} and D is the diagonal degree matrix with $d_{ii} = \sum_{j \in \mathcal{V}} w_{ij}$. The solution of (7.2) is provided by matrix T which contains the first k eigenvectors L_{sym} as columns.

As theoretically and empirically relevant to good performance, we reweight the affinities w_{ij}, as Galasso *et al.* (2014) suggest, by the number of fine superpixels to w^{Q}_{IJ} (cf. Galasso *et al.* (2014) for more details):

$$w^{Q}_{IJ} = \begin{cases} \sum_{i \in I} \sum_{j \in J} w_{ij} & \text{if } I \neq J, \\ \frac{1}{|I|} \sum_{i \in I} \sum_{j \in J} w_{ij} - \frac{(|I| - 1)}{|I|} \sum_{i \in I} \sum_{j \in \mathcal{V} \backslash I} w_{ij} & \text{if } I = J. \end{cases} \tag{7.3}$$

7.3.3 VSB100: Learning, Validating and Testing

Galasso *et al.* (2013) have introduced VSB100: a challenging video segmentation benchmark based on the HD quality videos from Sundberg *et al.* (2011), the boundary precision-recall (BPR) metric from Arbeláez *et al.* (2011) and a volume precision-recall metric (VPR) that reflects the properties of a good video segmentation, such as temporal consistency. Besides the PR curves, we report aggregate performance for BPR and VPR: optimal dataset scale [ODS], optimal segmentation scale [OSS], average precision [AP]. (We additionally report the length and number of clusters (NCL) statistics.)

The 100 videos are arranged into train (40) and test (60) set. We further split the training set into a *training* and *validation* sets, where 24 video sequences are used for learning the classifier and 16 videos are used for validation of the parameters. We compare with other approaches on the whole test set.

7.4 GRAPH CONSTRUCTION

Here we discuss the proposed graph construction $\mathcal{L}(\mathcal{G})$. First we consider learning for estimating the edge weights and the importance of topology in the setup of different classifiers. Then we consider calibration of classifier scores based on their reliability. Finally, we discuss edge selection, for a sparse efficient graph. We conduct these experiments on the training and validation sets.

7.4.1 Learning superpixel affinities

Let us consider the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, as introduced in Section 7.3, composed of superpixel nodes, connected over their spatio-temporal neighborhoods.

We propose the use of a classifier to learn the edge weights. To this end, we harvest from the training set pairs of superpixels connected by an affinity and provide them to a classifier along with their ground truth labelling (the indication whether two superpixels belong to the same video segment or not). Random Forest is used for learning.

There are four superpixel edge types: within, across 1, across 2 and across > 2 frames. While a single classifier should suffice for all, in our first experiments it turned out that its performance is extremely poor. By contrast, the use of multiple classifiers is beneficial. We attribute this to data unbalance (the edges within frames are the vast majority) and to scarcity of training samples (esp. compared to the large image and video variability).

We set therefore to consider four classifiers for the four edge types. The corresponding available features are:

within frame: sta, sta_{χ^2}, stm, stm_{χ^2}, aba, abm, sd, $text$, $text_{\chi^2}$, $size$;

across 1: sta, sta_{χ^2}, stm, stm_{χ^2}, stt, sd, $text$, $text_{\chi^2}$, $size$;

across 2: ltt, cit, stt, sd, $text$, $text_{\chi^2}$, $size$;

across > 2: ltt, cit, sd, td, $size$.

In our experience the Random Forest classifier profits from removing redundant or irrelevant features. Therefore for each affinity type we validate the subset of features to improve the model. The maximum set which we consider consists of 10 features (within frame), therefore we can test each possible combination finding the one which maximizes the average precision of the classifier. This is an exhaustive search of the feature space; however, in this particular setting it is computationally tractable as the feature set is quite small. We train a new classifier for each subset of features and validate the performance on a subset of the validation set. The best performing feature sets for each affinity type are reported in Table 7.1. Our findings on the importance of each feature for each affinity type are in agreement with Galasso *et al.* (2012) (the most contributive are highlighted in bold in the table).

Our experiments confirm that only considering pairs of superpixels in the training set which have at least 60% overlap with ground truth objects improves results, as also noted in Oneata *et al.* (2014); Raza *et al.* (2013). Further stricter thresholds do not benefit the performance and also reduce the number of training samples.

In Figure 7.2, we plot precision-recall curves comparing our learnt affinities against the original ones of Galasso *et al.* (2012), for which weighted-product combinations of motion, appearance and shape features were hand-tuned. Note that the improvement of our curve (red) is particularly prominent at the high-precision regimes. High precision scores are important as they correspond to decisions taken with most confidence, thus most detrimental to the graph partitioning when wrong.

Implementation details. We use the Random Forest implementation of Dollár. The number of features to sample for each node split is set to $\sqrt{F}$, where F is the dimensionality of the feature space. As weak learners we use binary split functions, and the maximum tree depth is set to 50. Split thresholds are chosen to optimize the Gini impurity. The minimum number of data points required to split a node in the tree is set to 15. Ensemble averaging is used to fuse the predictions of trees. Other parameters, such as number of trees [250, 350, 150, 300] and minimum number of data points allowed at leaves [10, 15, 5, 15] are validated on the subset of the validation data and differs for each affinity type, depending on the dimensionality of training sets.

7.4.2 Topology of the graph

Note from Figure 7.2 the overall performance (red curves) of the affinities learnt for the across 1 (Figure 7.2(b)) and the across 2 type (Figure 7.2(c)). The across 1 type have 55% precision (we take the overall precision at 100% recall). These have therefore 55% chance of correctness compared to 82% of the across 2 type learnt affinities. The across 1 affinities should ideally be more accurate, as they connect superpixels closer in time.

Let us take a closer look at the graph topology of Galasso *et al.* (2014), i.e. the edge connectivity $\mathcal{E}$. In the case of connectivity between superpixels within or across 1 frame, the graph is densified by using edges among neighboring superpixels (we call these *layer-1 neighbors*) and among more distant superpixels which share the same neighbor (we name these *layer-2 neighbors*). While the across 1 type affinities consider both direct temporal neighbors (best temporally-matching superpixel edges,

Affinity type	Set of features
i. within frame	{**sta**, sta_{χ^2}, **stm**, stm_{χ^2}, **aba**, *abm*, *sd*, *text*, $text_{\chi^2}$, *size*}
ii. across 1 frame	{**sta**, sta_{χ^2}, *stm*, **stt**, **sd**, *text*, $text_{\chi^2}$, *size*}
iii. across 2 frames	{**ltt**, **cit**, **stt**, *sd*, *size*}
iv. across $>$ 2 frames	{**ltt**, **cit**, *sd*, *td*}

Table 7.1: Set of features for learning.

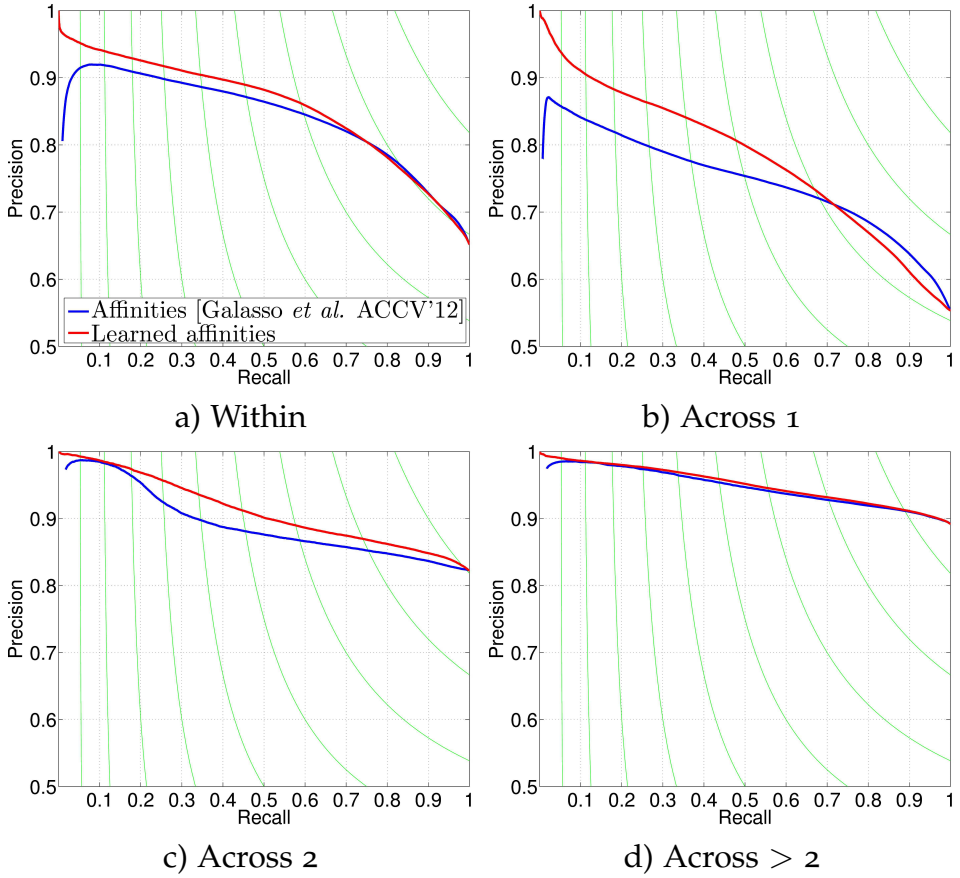


Figure 7.2: Affinity scores designed by Galasso *et al.* (2012) vs learned affinities.

according to optical flow propagation) and layer-2 neighbors, the across 2 type affinities only consider layer-1 neighbors.

We propose to treat the topologically different neighbors separately, which we illustrate in Figure 7.3, whereby we plot precision-recall curves for all types of our learnt affinities. We separate the two topologies both for the within and the across 1 type and re-learn separate classifiers. The results in Figure 7.3 show that now the layer-1 across 1 affinities reaches the overall performance (85%) of the across 2 affinity, and the corresponding performance of the within type also raises to 80%. As for the across 2 type, also the across > 2 type only has layer-1 neighbors and is therefore not affected by the topological procedure.

Taking into account the topology of the graph increases performance and improves the edge-selection procedure (cf. Section 7.4.4). Treating separately the two neighbor layers, video segmentation performance increases (on the validation set) by 2% on the BPR and 3% on the VPR measures of VSB100 (Galasso *et al.*, 2013) (cf. Figure 7.5). (These experiments are conducted by changing the topology of the graph and selecting edges with precision higher than 97% for all affinity types.)

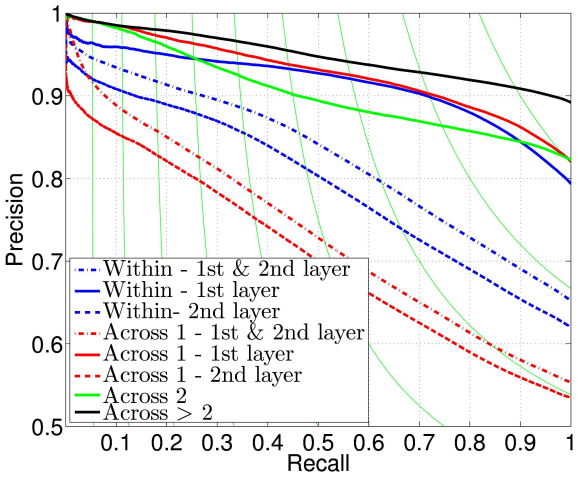


Figure 7.3: Performance of affinities, defined by the original graph topology (Galasso *et al.*, 2014).

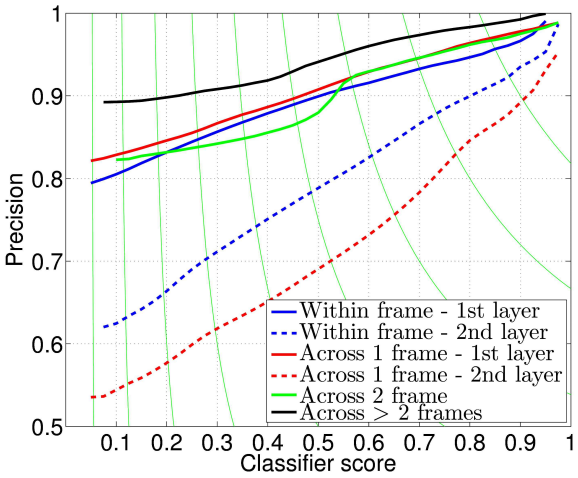


Figure 7.4: Calibration of classifier scores.

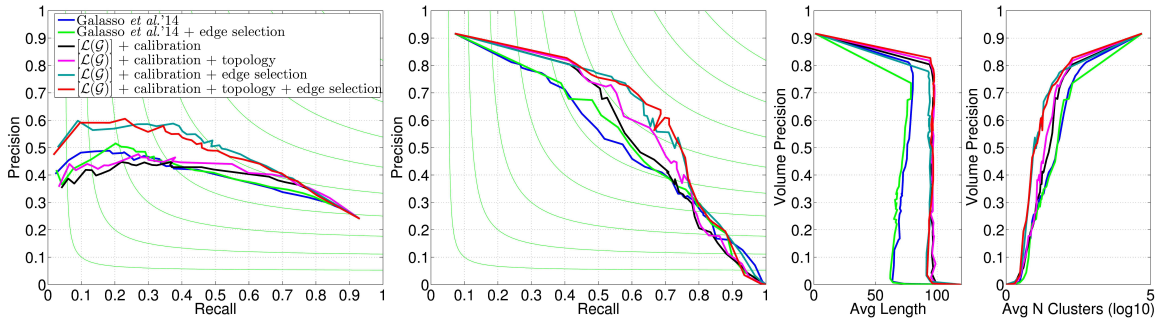
7.4.3 Calibration of classifier outputs

An ideal subsequent processing of the graph would be the selection of the most likely edges (assuming that these be correct) and the deletion of wrong ones. This is desirable because it sparsifies the graph and reduces the chance of segmentation errors. For this purpose the classifier scores should correspond to the confidence measure of two superpixels being merged together. However, the classifier outputs for different affinity types have different ranges and provide different confidence levels.

We propose a probabilistic interpretation of the learnt scores and to calibrate the classifier outputs based on their performance on the validation set. We define a linear mapping $\Pi : S \mapsto P$, such that the classifier score s is approximated by its precision value p. We mean by precision p the ratio of true positive edges among all weights higher than or equal s. Precision is taken as a proxy to the true posterior probability (affinity between two nodes).

For each affinity type we estimate its own calibration function, which is illustrated in Figure 7.4. This calibration is an easier interpretation of the classifier outputs and serves to align the scores to their quality. This is important when combining multiple classifiers, as also noted by Hallman and Fowlkes (2015).

The calibration of classifier outputs is not dependent on the choice of the learning algorithm. The proposed procedure provides a way to encode edge weights and in our experience can help to improve the clustering performance. The calibrated classifier output scores are used as edge weights in the graph.



BPR VPR Length NCL

	BPR			VPR			Length	NCL
Algorithm	ODS	OSS	AP	ODS	OSS	AP	$\mu(\delta)$	μ
Galasso *et al.* (2014)	0.46	0.49	0.37	0.54	0.61	0.56	75.94(46.39)	60
Galasso *et al.* (2014) + edge selection	0.47	0.51	0.37	0.57	0.62	0.57	74.14(46.26)	70
$[\mathcal{L}(\mathcal{G})]$ + calibration	0.49	0.53	0.36	0.59	0.65	0.59	96.83(34.42)	50
$[\mathcal{L}(\mathcal{G})]$ + calibration + topology	0.51	0.54	0.38	0.62	0.67	0.62	97.85(33.55)	60
$[\mathcal{L}(\mathcal{G})]$ + calibration + edge selection	**0.52**	0.57	**0.44**	0.65	**0.70**	0.64	94.23(37.59)	20
$[\mathcal{L}(\mathcal{G})]$ + calibr. + topology + edge select.	**0.52**	**0.58**	**0.44**	**0.66**	**0.70**	**0.65**	96.34(37.24)	20

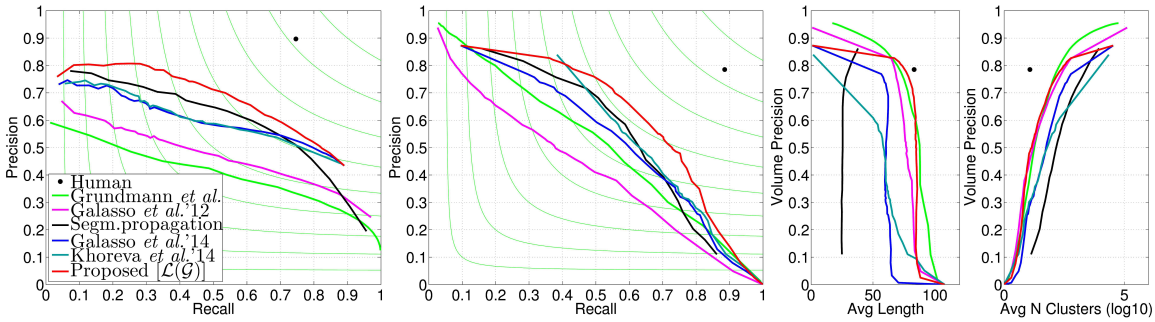
Figure 7.5: Comparison of the proposed graph learning method with the baseline algorithm of Galasso *et al.* (2014), on the validation set of VSB100 Galasso *et al.* (2013). The plots and table show BPR and VPR measures, aggregate performances ODS, OSS and AP, and length statistics (mean μ, std. δ, no. clusters NCL) (cf. Section 7.4.4 for details).

7.4.4 Edge selection

Following the argument of the previous section, we now modify the graph structure by reducing the number of edges and selecting the ones with high confidence. Each affinity type is thresholded with some confidence level, reducing the number of temporal and spatial edges in the graph. The goal is to have a connected graph with a minimal set of the most certain edges, as for maximal sparsity and the least chance of segmentation error.

For finding the optimal thresholds for each affinity type grid search is applied. We find the confidence levels for four affinity types which provide the best performance on the validation set. We measure the performance as the sum of F-measures (ODS, OSS) and AP for BPR and VPR metric. We restrict the candidate space of the thresholds for each affinity type to [0.5;1], as the goal is to leave the most confident edges which have at least 50% precision. Edge selection turns out to be essential for best performance, cf. the next discussion.

We also explored other procedures for edge selection, such as kNN, but they all underperform by large margins. Our edge selection produces a potentially unbalanced (nodes have different number of neighbors) but better graph.



BPR VPR Length NCL

	BPR			VPR			Length	NCL
Algorithm	ODS	OSS	AP	ODS	OSS	AP	$\mu(\delta)$	μ
Human	0.81	0.81	0.67	0.83	0.83	0.70	83.24(40.04)	12
Grundmann *et al.* (2010)	0.47	0.54	0.41	0.52	0.55	0.52	87.69(34.02)	19
Galasso *et al.* (2012)	0.51	0.56	0.45	0.45	0.51	0.42	80.17(37.56)	8
Galasso *et al.* (2013)	0.61	0.65	0.59	0.59	0.62	0.56	25.50(36.48)	258
Galasso *et al.* (2014)	0.62	0.66	0.54	0.55	0.59	0.55	61.25(40.87)	80
Khoreva *et al.* (2014)	0.61	0.66	0.52	0.58	0.61	0.58	51.72(39.90)	177
Proposed [$\mathcal{L}(\mathcal{G})$]	**0.64**	**0.70**	**0.61**	**0.63**	**0.66**	**0.63**	83.41(35.27)	50

Figure 7.6: Comparison of video segmentation algorithms with our proposed method on the test set of VSB100 Galasso *et al.* (2013) (cf. Section 7.5 for details).

7.4.5 Discussion

In Figure 7.5, we analyze how the learnt graph [$\mathcal{L}(\mathcal{G})$] and the proposed steps improve on the (validation) performance, with respect to the baseline algorithm of Galasso *et al.* (2014).

Given a learnt and calibrated graph (3rd row), topology improves 2.2% (4th row, average improvement over all six measures) while edge selection improves 5.2% (5th row). Edge selection is thus more important than topology. Adding topology on top of edge selection further contributes 0.5%. The importance of edge selection contrasts previous literature (Ren and Malik, 2003; Turaga *et al.*, 2009; Kim *et al.*, 2013), all concerned with edge weights.

To further test the importance of edge selection, we have applied this to the baseline algorithm of Galasso *et al.* (2014) (1st and 2nd rows). The improvement is only marginal (1.3%). We conclude therefore that a pre-requisite for successful edge selection is weight calibration plus the good performance of the classifier in the high precision regime (see Figure 7.2).

7.5 EXPERIMENTAL EVALUATION

In Figure 7.6 we compare the proposed method to the baseline (Galasso *et al.*, 2014) as well as video segmentation algorithms of Grundmann *et al.* (2010), Galasso *et al.*

	BPR			VPR		
Algorithm	ODS	OSS	AP	ODS	OSS	AP
Galasso *et al.* (2014) - 1–SC (Hein and Bühler, 2010)	0.61	0.64	0.52	0.55	0.60	0.54
$[\mathcal{L}(\mathcal{G})]$ - 1–SC (Hein and Bühler, 2010)	**0.63**	**0.69**	**0.63**	**0.60**	**0.65**	**0.59**
Galasso *et al.* (2014) - GRACLUS (Dhillon *et al.*, 2007)	0.59	0.64	0.51	0.34	0.46	0.31
$[\mathcal{L}(\mathcal{G})]$ - GRACLUS (Dhillon *et al.*, 2007)	**0.62**	**0.67**	**0.52**	**0.54**	**0.60**	**0.53**
Galasso *et al.* (2014) - MCL (van Dongen, 2008)	0.59	0.64	**0.45**	0.40	0.46	0.37
$[\mathcal{L}(\mathcal{G})]$ - MCL (van Dongen, 2008)	**0.64**	**0.68**	0.39	**0.58**	**0.59**	**0.59**

Table 7.2: General applicability of the proposed graph construction. We have tested different clustering methods with the graph of Galasso *et al.* (2014) and our learnt graph. In all cases the learnt graph yields better performance and thus generalizes beyond the employed spectral clustering.

(2012, 2013) and our approach proposed in Chapter 6 (Khoreva *et al.*, 2014) on the test set of VSB100 (Galasso *et al.*, 2013). We consider the graph $[\mathcal{L}(\mathcal{G})]$ with the learnt topology and edge weights proposed in Section 7.4.

The proposed method improves the performance of Galasso *et al.* (2014) on both BPR and VPR by a large margin, as it appears both in the plots and the tables (average improvement of 4% in BPR and 8% in VPR, 6% on all measures). We outperform all considered video segmentation algorithms and the challenging segmentation propagation baseline (Galasso *et al.*, 2013).

The proposed graph construction, however, is directly applicable to other graph-based techniques. We have tested different graph partitioning methods (Bühler and Hein, 2009; Hein and Bühler, 2010; Dhillon *et al.*, 2007; van Dongen, 2008) with the graph of Galasso *et al.* (2014) and our learnt graph, the results are presented in Table 7.2. For all three tested methods our learnt graph improves significantly the performance both on BPR and VPR (up to 6–10% on average). This shows that our graph construction generalizes beyond the employed spectral clustering technique. Note that the 1-spectral clustering approach (Bühler and Hein, 2009; Hein and Bühler, 2010) outperforms spectral clustering in terms of AP with respect to BPR while being worse on VPR.

Regarding runtime, the efficiency of the algorithm depends on the number of superpixels n (nodes in the graph). The (test-time) Random Forests classification runtime is negligible with respect to feature computation and graph partition. In spectral clustering, the bottleneck is the eigendecomposition: the Lanczos method has complexity $O(kE)$ and the iteration number scales with $\sim \log E$ (k the number of eigenvectors and E the number of edges in the graph, which scales linearly with n, approx. $\sim 366n$). In our graph due to the edge selection procedure the average number of edges is reduced to 15% and the constructed graph is much sparser, hence the reduction in runtime of 55% with respect to Galasso *et al.* (2014). E.g. runtime of "soccer" reduces from 4.8 min to 2.9 min, "hippo fight" from 9.3 min to 4.4 min.

We illustrate qualitative results, comparing in Figure 7.7 our proposed algorithm to other video segmentation methods (Grundmann *et al.*, 2010; Galasso *et al.*, 2012,

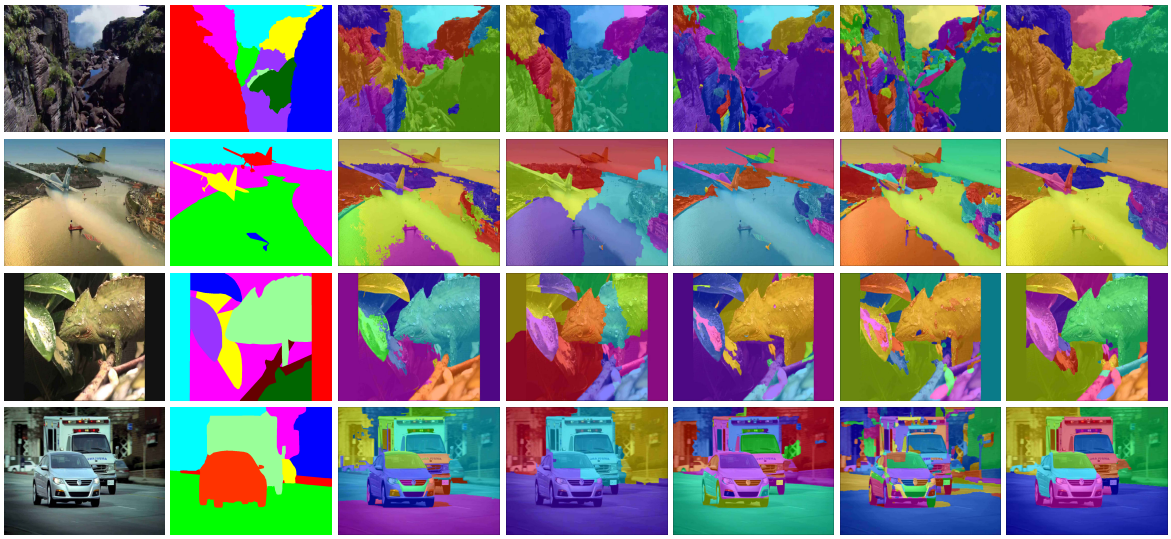


Figure 7.7: Comparison of video segmentation results of algorithms (Grundmann *et al.*, 2010; Galasso *et al.*, 2012, 2013, 2014) and our proposed method $[\mathcal{L}(\mathcal{G})]$ to one of ground truths (Galasso *et al.*, 2013). We report for each algorithm the coarse-to-fine segmentation level with best performance in VPR. Our approach qualitatively improves on the algorithm of Galasso *et al.* (2014), better discriminating visual objects with less number of clusters (cf. Section 7.5 for details).

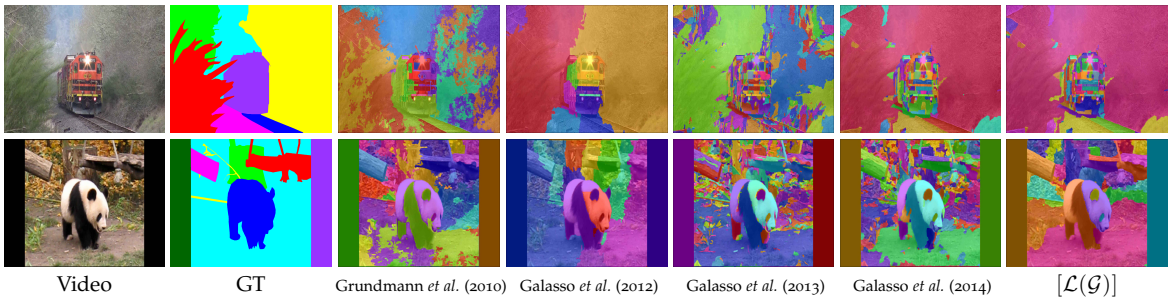


Figure 7.8: Failure cases for the algorithms (Grundmann *et al.*, 2010; Galasso *et al.*, 2012, 2013, 2014) and the proposed graph learning method $[\mathcal{L}(\mathcal{G})]$. All methods fail to correctly discern objects, oversegmenting the foreground and background due to the misleading appearance differences and textured background.

2013, 2014). Figure 7.7 supports the positive quantitative results. The proposed approach allows to better distinguish visual objects with well-localized boundaries and limited label leakage. Segmentations provided by our method capture better motion and appearance self-contained within the objects, distinguishing the homogeneous areas of foreground and background with less number of clusters. However, a failure cases show further potential for improvement (see Figure 7.8).

7.6 CONCLUSIONS

In this chapter we addressed the classifier based graph construction procedure for video segmentation. We proposed an empirical approach to learn both the edge topology and weights of the graph. While combining well-established features by means of a classifier and calibrating the classifier scores by its accuracy we alter the graph structure selecting the most confident edges. Our method of learning the graph helps to improve both performance on the challenging VSB100 benchmark as well as efficiency without changing the graph partitioning model.

In the next chapter we aim to improve the graph nodes - superpixels, which are the starting point for unary and pairwise terms, and thus have a direct influence on the final quality of video segmentation techniques proposed in the current and previous chapters.

8 IMPROVED IMAGE BOUNDARIES FOR BETTER VIDEO SEGMENTATION

While in the two previous chapters we have focused on the construction of the graph as well as solving the graph partitioning problem with must-link constraints, this chapter focuses on better superpixels for video segmentation.

We demonstrate by a comparative analysis that superpixels extracted from boundaries perform best, and show that boundary estimation can be significantly improved via image and time domain cues. With superpixels generated from our better boundaries we observe consistent improvement for two video segmentation methods in two different datasets.

8.1 INTRODUCTION

Class-agnostic image and video segmentation have shown to be helpful in diverse computer vision tasks such as object detection (via object proposals) (Krähenbühl and Koltun, 2014; Pont-Tuset *et al.*, 2016; Humayun *et al.*, 2014, 2015), semantic video segmentation (as pre-segmentation) (Dai *et al.*, 2015b), activity recognition (by computing features on voxels) (Taralova *et al.*, 2014), or scene understanding (Jain *et al.*, 2013).

Both image and video segmentation have seen steady progress recently leveraging advanced machine learning techniques. A popular and successful approach consists of modeling segmentation as a graph partitioning problem (Fragkiadaki and Shi, 2012; Ochs *et al.*, 2014; Keuper and Brox, 2016), where the nodes represent pixels or superpixels, and the edges encode the spatio-temporal structure. Previous work focused on solving the partitioning problem (Brox and Malik, 2010; Grundmann *et al.*, 2010; Palou and Salembier, 2013; Yi and Pavlovic, 2015), on the unary and pairwise terms of the graph (Galasso *et al.*, 2014) and on the graph construction itself (Ren and Malik, 2003; Turaga *et al.*, 2009; Khoreva *et al.*, 2015).

The aim of this work is to improve video segmentation by focusing on the graph nodes themselves, the video superpixels. These nodes are the starting point for unary and pairwise terms, and thus directly impact the final segmentation quality. Good superpixels for video segmentation should both be temporally consistent and give high boundary recall, and, in the case of graph-based video segmentation, for efficient runtime should enable to use a few superpixels per frame which is related to high boundary precision.

Our experiments show that existing classical superpixel/voxel methods (Chang *et al.*, 2013; Achanta *et al.*, 2012; Bergh *et al.*, 2013) underperform for graph-based

Video | TSP superpixels (Chang *et al.*, 2013) (117 spx) | $gPb_{\mathcal{I}}$ superpixels (Arbeláez *et al.*, 2011) (101 spx) | Our superpixels (101 spx)

Figure 8.1: Graph based video segmentation relies on having high quality superpixels/voxels as starting point (graph nodes). We explore diverse techniques to improve boundary estimates, which result in better superpixels, which in turn has a significant impact on final video segmentation.

video segmentation and superpixels built from per-frame boundary estimates are more effective for the task (see Section 8.5). We show that boundary estimates can be improved when using image cues combined with object-level cues, and by merging with temporal cues. By fusing image and time domain cues, we can significantly enhance boundary estimation in video frames, improve per-frame superpixels, and thus improve video segmentation.

In particular we contribute:

- A comparative evaluation of the importance of the initial superpixels/voxels for graph-based video segmentations (Section 8.5).
- Significantly improved boundary estimates (and thus per-frame superpixels) by the careful fusion of image (Section 8.6.1) and time (Section 8.6.2) domain cues.
- The integration of high-level object-related cues into the local image segmentation processing (Section 8.6.1.1).
- High-quality video segmentation results on the VSB100 (Galasso *et al.*, 2013) and BMDS (Brox and Malik, 2010) datasets.

8.2 PREVIOUS WORK ON SUPERPIXELS/VOXELS

Video segmentation can be seen as a clustering problem in the 3D spatial-temporal volume. Considering superpixels/voxels as nodes, graphs are a natural way to address video segmentation and there are plenty of approaches to process the graphs.

Previous work covers various aspects related to graph based video segmentation. Several papers have addressed the features for video segmentation (Brox and Malik, 2010; Grundmann *et al.*, 2010; Palou and Salembier, 2013) and some work has

addressed the graph construction (Ren and Malik, 2003; Turaga *et al.*, 2009). While these methods are based on superpixels none of them examines the quality of the respective superpixels for graph-based video segmentation. To the best of our knowledge, this work is the first to thoroughly analyze and advance superpixel methods in the context of video segmentation.

We distinguish two groups of superpixel methods. The first one is the classical superpixel/voxel methods (Chang *et al.*, 2013; Achanta *et al.*, 2012; Bergh *et al.*, 2013; Levinshtein *et al.*, 2009). These methods are designed to extract superpixels of homogeneous shape and size, in order for them to have a regular topology. Having a regular superpixel topology has shown a good basis for image and video segmentation (Grundmann *et al.*, 2010; Papazoglou and Ferrari, 2013; Badrinarayanan *et al.*, 2013; Ren and Malik, 2003).

The second group are based on boundary estimation and focus on the image content. They extract superpixels by building a hierarchical image segmentation (Arbeláez *et al.*, 2011; Isola *et al.*, 2014; Dollár and Zitnick, 2015; Pont-Tuset *et al.*, 2016) and selecting one level in the hierarchy. These methods generate superpixels of heterogeneous size, that are typically fairly accurate on each frame but may jitter over time. Superpixels based on per-frame boundary estimation are employed in many state-of-the-art video segmentation methods (Galasso *et al.*, 2014; Vazquez-Reina *et al.*, 2010; Jain *et al.*, 2013; Yi and Pavlovic, 2015).

In this work we argue that boundaries based superpixels are more suitable for graph-based video segmentation, and propose to improve the extracted superpixels by exploring temporal information such as optical flow and temporal smoothing.

8.3 VIDEO SEGMENTATION METHODS

For our experiments we consider two open source graph-based video segmentation methods (Galasso *et al.*, 2013, 2014). Both of them rely on superpixels extracted from hierarchical image segmentation (Arbeláez *et al.*, 2011), which we aim to improve.

Spectral graph reduction (Galasso *et al.*, 2014). Our first baseline is composed of three main parts:

1. *Extraction of superpixels.* Superpixels are image-based pixel groupings which are similar in terms of colour and texture, extracted by using the state-of-the-art image segmentation of Arbeláez *et al.* (2011). These superpixels are accurate but not temporally consistent, as only extracted per frame.

2. *Feature computation.* Superpixels are compared to their (spatio-temporal) neighbors and affinities are computed between pairs of them based on appearance, motion and long term point trajectories (Ochs *et al.*, 2014), depending on the type of neighbourhood (e.g. within a frame, across frames, etc.).

3. *Graph partitioning.* Video segmentation is cast as the grouping of superpixels into video volumes. Galasso *et al.* (2014) employ either a spectral clustering or

normalised cut formulation for incorporating a reweighing scheme to improve the performance.

In our work we focus on the first part. We show that superpixels extracted from stronger boundary estimation help to achieve better segmentation performance without altering the underlying features or the graph partitioning method.

Segmentation propagation (Galasso *et al.*, 2013). As the second video segmentation method we consider the baseline proposed in Galasso *et al.* (2013). This method does greedy matching of superpixels by propagating them over time via optical flow. This "simple" method obtains good results on VSB100. We therefore also report how superpixels extracted via hierarchical image segmentation based on our proposed boundary estimation improve this baseline.

8.4 VIDEO SEGMENTATION EVALUATION

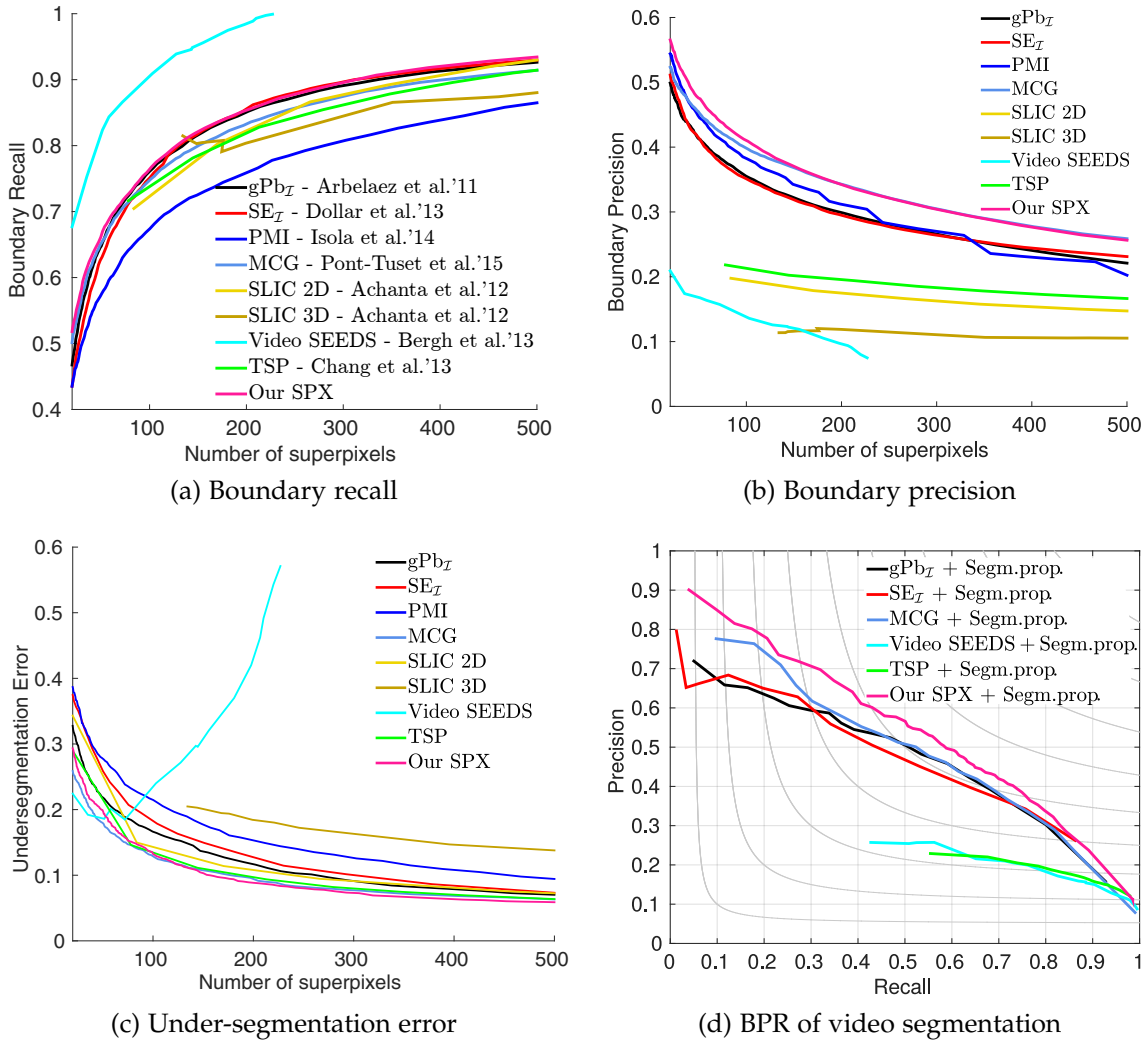
VSB100. We consider for learning and for evaluation the challenging video segmentation benchmark VSB100 (Galasso *et al.*, 2013) based on the HD quality video sequences of Sundberg *et al.* (2011), containing natural scenes as well as motion pictures, with heterogeneous appearance and motion. The dataset is arranged into train (40 videos) and test (60) set. Additionally we split the training set into a training (24) and validation set (16).

The evaluation in VSB100 is mainly given by:

Precision-recall plots (BPR, VPR): VSB100 distinguishes a boundary precision-recall metric (BPR), measuring the per-frame boundary alignment between a video segmentation solution and the human annotations, and a volume precision-recall metric (VPR), reflecting the temporal consistency of the video segmentation result.

Aggregate performance measures (AP, ODS, OSS): for both BPR and VPR, VSB100 reports average precision (AP), the area under the precision-recall curves, and two F-measures where one is measured at an optimal dataset scale (ODS) and the other at an optimal segmentation scale (OSS) (where "optimal" stands for oracle provided).

BMDS. To show the generalization of the proposed method we further consider the Berkeley Motion Segmentation Dataset (BMDS) (Brox and Malik, 2010), which consists of 26 VGA-quality videos, representing mainly humans and cars. Following prior work (Khoreva *et al.*, 2014) we use 10 videos for training and 16 as a test set, and restrict all video sequences to the first 30 frames.



(a) Boundary recall

(b) Boundary precision

(c) Under-segmentation error

(d) BPR of video segmentation

Figure 8.2: Comparison of different superpixel/voxel methods, and their use for video segmentation. VSB100 validation set. SPX: superpixels. Segm. prop.: segmentation propagation (Galasso *et al.*, 2013) (see Section 8.3).

8.5 SUPERPIXELS AND SUPERVOXELS

Graph-based video segmentation methods rely on superpixels to compute features and affinities. Employing superpixels as pre-processing stage for video segmentation provides a desirable computational reduction and a powerful per-frame representation. Ideally these superpixels have high boundary recall (since one cannot recover from missing recall), good temporal consistency (to make matching across time easier), and are as few as possible (in order to reduce the chances of segmentation errors; to accelerate overall computation and reduce memory needs).

In this section we explore which type of superpixels are most suitable for graph-based video segmentation.

Superpixel/voxel methods. Many superpixel/voxel methods have been explored in the past. We consider the most promising ones in the experiments of Figure 8.2. SLIC 2D/3D (Achanta *et al.*, 2012) is a classic method to obtain superpixels via iterative clustering (in space and space-time domain). TSP (Chang *et al.*, 2013) extends SLIC to explicitly model temporal dynamics. Video SEEDS (Bergh *et al.*, 2013) is similar to SLIC 3D, but uses an alternative optimization strategy. Other than classic superpixel/voxel methods we also consider superpixels generated from per-frame hierarchical segmentation based on boundary detection (ultrametric contour maps (Arbeláez *et al.*, 2011)). We include $gPb_{\mathcal{I}}$ (Arbeláez *et al.*, 2011), $SE_{\mathcal{I}}$ (Dollár and Zitnick, 2015), PMI (Isola *et al.*, 2014) and MCG (Pont-Tuset *et al.*, 2016) as sources of boundary estimates.

Superpixel evaluation. We compare superpixels by evaluating the recall and precision of boundaries and the under-segmentation error (Neubert and Protzel, 2013) as functions of the average number of superpixels per frame. We also use some of them directly for video segmentation (Figure 8.2d). We evaluate (use) all methods on a frame by frame basis; supervoxel methods are expected to provide more temporally consistent segmentations than superpixel methods.

Results. Boundary recall (Figure 8.2a) is comparable for most methods. Video SEEDS is an outlier, showing very high recall, but low boundary precision (8.2b) and high under-segmentation error (Figure 8.2c). $gPb_{\mathcal{I}}$ and $SE_{\mathcal{I}}$ reach the highest boundary recall with fewer superpixels. Per-frame boundaries based superpixels perform better than classical superpixel methods on boundary precision (Figure 8.2b). From these figures one can see the conflicting goals of having high boundary recall, high precision, and few superpixels.

We additionally evaluate the superpixel methods using a region-based metric: under-segmentation error (Neubert and Protzel, 2013). Similar to the boundary results, the curves are clustered in two groups: TSP-like and $gPb_{\mathcal{I}}$-like quality methods, where the latter underperform due to the heterogeneous shape and size of superpixels (Figure 8.2c).

Figure 8.2d shows the impact of superpixels for video segmentation using the baseline method of Galasso *et al.* (2013). We pick TSP as a representative superpixel method (fair quality on all metrics), Video SEEDS as an interesting case (good boundary recall, bad precision), $SE_{\mathcal{I}}$ and MCG as good boundary estimation methods, and the baseline $gPb_{\mathcal{I}}$ (used in Galasso *et al.* (2013)). Although classical superpixel methods have lower under-segmentation error than boundaries based superpixels, when applied for video segmentation the former underperform (both on boundary and volume metrics), as seen in Figure 8.2d. Boundary quality measures seem to be a good proxy to predict the quality of superpixels for video segmentation. Both in boundary precision and recall metrics having stronger initial superpixels leads to better results.

Intuition. Figure 8.1 shows a visual comparison of TSP superpixels versus $gPb_{\mathcal{I}}$ superpixels (both generated with a similar number of superpixels). By design, most classical superpixel methods have a tendency to generate superpixels of comparable size. When requested to generate fewer superpixels, they need to trade-off quality versus regular size. Methods based on hierarchical segmentation (such as $gPb_{\mathcal{I}}$) generate superpixels of heterogeneous sizes and more likely to form semantic regions. For a comparable number of superpixels techniques based on image segmentation have more freedom to provide better superpixels for graph-based video segmentation than classical superpixel methods.

Conclusion. Based both on quality metrics and on their direct usage for graph-based video segmentation, boundary based superpixels extracted via hierarchical segmentation are more effective than the classical superpixel methods in the context of video segmentation. The hierarchical segmentation is fully defined by the estimated boundary probability, thus better boundaries lead to better superpixels, which in turn has a significant impact on final video segmentation. In the next sections we discuss how to improve boundary estimation for video.

8.6 IMPROVING IMAGE BOUNDARIES

To improve the boundary based superpixels fed into video segmentation we seek to make best use of the information available on the videos. We first improve boundary estimates using each image frame separately (Section 8.6.1) and then consider the temporal dimension (Section 8.6.2).

8.6.1 Image domain cues

A classic boundary estimation method (often used in video segmentation) is $gPb_{\mathcal{I}}$ (Arbeláez *et al.*, 2011) ($\mathcal{I}$: image domain), we use it as a reference point for boundary quality metrics. In our approach we propose to use $SE_{\mathcal{I}}$ ("structured edges") (Dollár and Zitnick, 2015). We also considered the convnet based boundary detector of Xie and Tu (2015). However, employing boundaries of Xie and Tu (2015) to close the contours and construct per-frame hierarchical segmentation results in the performance similar to $SE_{\mathcal{I}}$ and significantly longer training time. Therefore in our system we employ $SE_{\mathcal{I}}$ due to its speed and good quality.

8.6.1.1 *Object proposals*

Methods such as $gPb_{\mathcal{I}}$ and $SE_{\mathcal{I}}$ use bottom-up information even though boundaries annotated by humans in benchmarks such as BSDS500 or VSB100 often follow object boundaries. In other words, an oracle having access to ground truth semantic object boundaries should allow to improve boundary estimation (in particular on the low recall region of the BPR curves). Based on this intuition we consider using

segment-level object proposal (OP) methods to improve initial boundary estimates ($\text{SE}_{\mathcal{I}}$). Object proposal methods (Krähenbühl and Koltun, 2014; Pont-Tuset *et al.*, 2016; Humayun *et al.*, 2014, 2015) aim at generating a set of candidate segments likely to have high overlap with true objects. Typically such methods reach $\sim$80% object recall with 10^3 proposals per image.

Based on initial experiments we found that the following simple approach obtains good boundary estimation results in practice. Given a set of object proposal segments generated from an initial boundary estimate, we average the contours of each segment. Pixels that are boundaries to many object proposals will have high probability of boundary; pixels rarely members of a proposal boundary will have low probability. With this approach, the better the object proposals, the closer we are to the mentioned oracle case.

We evaluated multiple proposals methods (Krähenbühl and Koltun, 2014; Pont-Tuset *et al.*, 2016; Humayun *et al.*, 2014) and found RIGOR (Humayun *et al.*, 2014) to be most effective for this use (Section 8.6.1.5). To the best of our knowledge this is the first time an object proposal method is used to improve boundary estimation. We name the resulting boundary map $\text{OP}\left(\text{SE}_{\mathcal{I}}\right)$.

8.6.1.2 *Globalized probability of boundary*

A key ingredient of the classic $\text{gPb}_{\mathcal{I}}$ (Arbeláez *et al.*, 2011) method consists on "globalizing boundaries". The most salient boundaries are highlighted by computing a weighted sum of the spatial derivatives of the first few eigenvectors of an affinity matrix built based on an input probability of boundary. The affinity matrix can be built either at the pixel or superpixel level. The resulting boundaries are named "spectral" probability of boundary, $\text{sPb}\left(\cdot\right)$. We employ the fast implementation from Pont-Tuset *et al.* (2016).

Albeit well known, such a globalization step is not considered by the latest work on boundary estimation (e.g. Dollár and Zitnick (2015); Bertasius *et al.* (2015a)). Since we compute boundaries at a single-scale, $\text{sPb}\left(\text{SE}_{\mathcal{I}}\right)$ is comparable to the SCG results in Pont-Tuset *et al.* (2016).

8.6.1.3 *Re-training*

Methods such as $\text{SE}_{\mathcal{I}}$ are trained and tuned for the BSDS500 image segmentation dataset (Arbeláez *et al.*, 2011). Given that VSB100 (Galasso *et al.*, 2013) is larger and arguably more relevant to the video segmentation task than BSDS500, we retrain $\text{SE}_{\mathcal{I}}$ (and RIGOR) for this task. In the following sections we report results of our system trained over BSDS500, or with VSB100. We will also consider using input data other than an RGB image (Section 8.6.2.1).

8.6.1.4 *Merging cues*

After obtaining complementary probabilities of boundary maps (e.g. $\text{OP}\left(\text{SE}_{\mathcal{I}}\right)$, $\text{sPb}\left(\text{SE}_{\mathcal{I}}\right)$, etc.), we want to combine them effectively. Naive averaging is inadequate

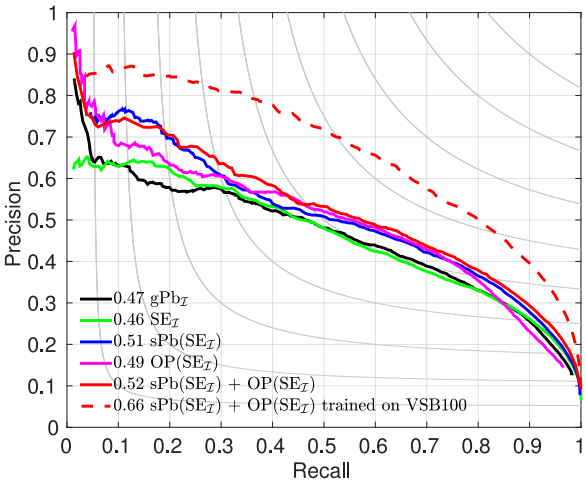


Figure 8.3: Progress when integrating various image domain cues (Section 8.6.1) in terms of BPR on VSB100 validation set.

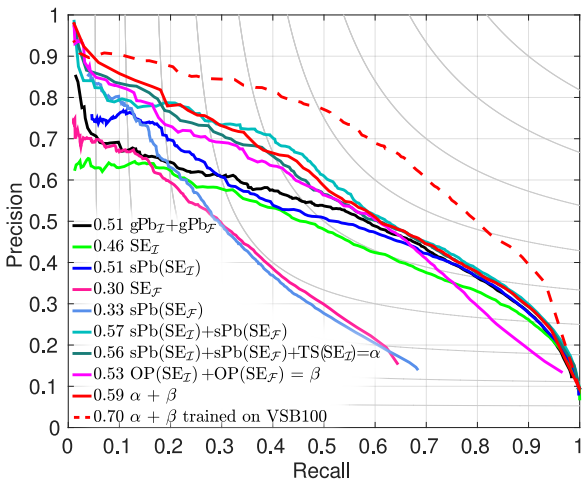


Figure 8.4: Progress when integrating image and time domain cues (Section 8.6.2) in terms of BPR on VSB100 validation set.

because boundaries estimated by different methods do not have pixel-perfect alignment amongst each other. Pixel-wise averaging or maxing leads to undesirable double edges (negatively affecting boundary precision).

To solve this issue we use the grouping technique from Pont-Tuset *et al.* (2016) which proposes to first convert the boundary estimate into a hierarchical segmentation, and then to align the segments from different methods. Note that we do not use the multi-scale part of Pont-Tuset *et al.* (2016). Unless otherwise specified all cues are averaged with equal weight. We use the sign "+" to indicate such merges.

8.6.1.5 *Boundary results when using image domain cues*

Figure 8.3 reports results when using the different image domain cues, evaluated over the VSB100 validation set. The gPb$_\mathcal{I}$ baseline obtains 47% AP, while SE$_\mathcal{I}$ (trained on BSDS500) obtains 46%. Interestingly, boundaries based on object proposals OP (SE$_\mathcal{I}$) from RIGOR obtain a competitive 49%, and, as expected, provide most gain in the high precision region of BPR. Globalization sPb (SE$_\mathcal{I}$) improves results to 51% providing a homogeneous gain across the full recall range. Combining sPb (SE$_\mathcal{I}$) and OP (SE$_\mathcal{I}$) obtains 52%. After retraining SE$_\mathcal{I}$ on VSB100 we obtain our best result of 66% AP (note that all cues are affected by re-training SE$_\mathcal{I}$).

Conclusion. Even when using only image domain cues, large gains can be obtained over the standard gPb$_\mathcal{I}$ baseline.

8.6.2 Temporal cues

The results of Section 8.6.1 ignore the fact that we are processing a video sequence. In the next sections we describe two different strategies to exploit the temporal

dimension.

8.6.2.1 *Optical flow*

We propose to improve boundaries for video by employing optical flow cues. We use the state-of-the-art EpicFlow (Revaud *et al.*, 2015) algorithm, which we feed with our $\text{SE}_{\mathcal{I}}$ boundary estimates.

Since optical flow is expected to be smooth across time, if boundaries are influenced by flow, they will become more temporally consistent. Our strategy consists of computing boundaries directly over the forward and backward flow map, by applying SE over the optical flow magnitude (similar to one of the cues used in Fragkiadaki *et al.* (2015)). We name the resulting boundaries map $\text{SE}_{\mathcal{F}}$ ($\mathcal{F}$: optical flow). Although the flow magnitude disregards the orientation information from the flow map, in practice discontinuities in magnitude are related to changes in flow direction.

We then treat $\text{SE}_{\mathcal{F}}$ similarly to $\text{SE}_{\mathcal{I}}$ and compute OP ($\text{SE}_{\mathcal{F}}$) and sPb ($\text{SE}_{\mathcal{F}}$) over it. All these cues are finally merged using the method described in Section 8.6.1.4.

8.6.2.2 *Time smoothing*

The goal of our new boundaries based superpixels is not only high recall, but also good temporal consistency across frames. A naive way to improve temporal smoothness of boundaries consists of averaging boundary maps of different frames over a sliding window; differences across frames would be smoothed out, but at the same time double edge artefacts (due to motion) would appear (reduced precision).

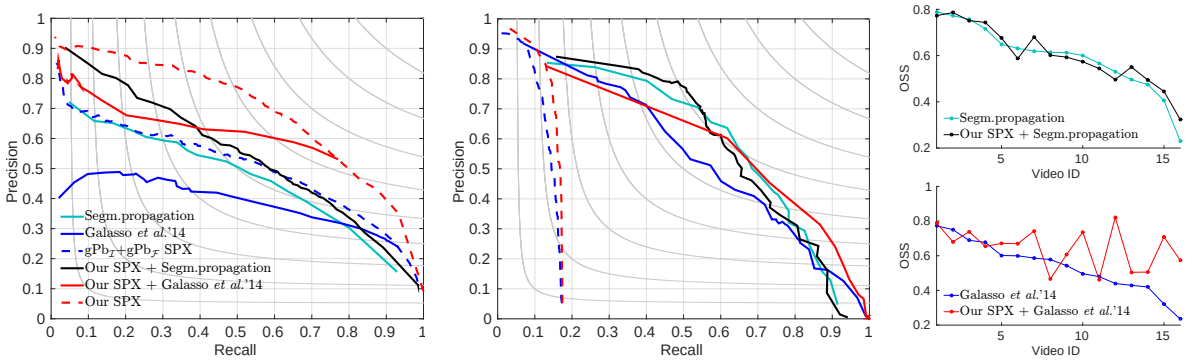
We propose to improve temporal consistency by doing a sliding window average across boundary maps of several adjacent frames. For each frame t, instead of naively transferring boundary estimates from one frame to the next, we warp frames $t_{\pm i}$ using optical flow with respect to frame t; thus reducing double edge artefacts. For each frame t we treat warped boundaries from frames $t_{\pm i}$ as additional cues, and merge them using the same mechanism as in Section 8.6.1.4. This merging mechanism is suitable to further reduce the double edges issue.

8.6.2.3 *Boundary results when using temporal cues*

The curves of Figure 8.4 show the improvement gained from optical flow and temporal smoothing.

Optical flow. Figure 8.4 shows that on its own flow boundaries are rather weak ($\text{SE}_{\mathcal{F}}$, sPb ($\text{SE}_{\mathcal{F}}$)), but they are quite complementary to image domain cues (sPb ($\text{SE}_{\mathcal{I}}$) versus sPb ($\text{SE}_{\mathcal{I}}$)+sPb ($\text{SE}_{\mathcal{F}}$)).

Temporal smoothing. Using temporal smoothing (sPb ($\text{SE}_{\mathcal{I}}$)+sPb ($\text{SE}_{\mathcal{F}}$)+TS ($\text{SE}_{\mathcal{I}}$) $=\alpha$) leads to a minor drop in boundary precision, in comparison with sPb ($\text{SE}_{\mathcal{I}}$)+ sPb ($\text{SE}_{\mathcal{F}}$) in Figure 8.4. It should be noted that there is an inherent tension between



(a) BPR on validation set (b) VPR on validation set (c) OSS per video

Figure 8.5: VSB100 validation set results of different video segmentation methods. Dashed lines indicate only frame-by-frame processing (see Section 8.7.1 for details).

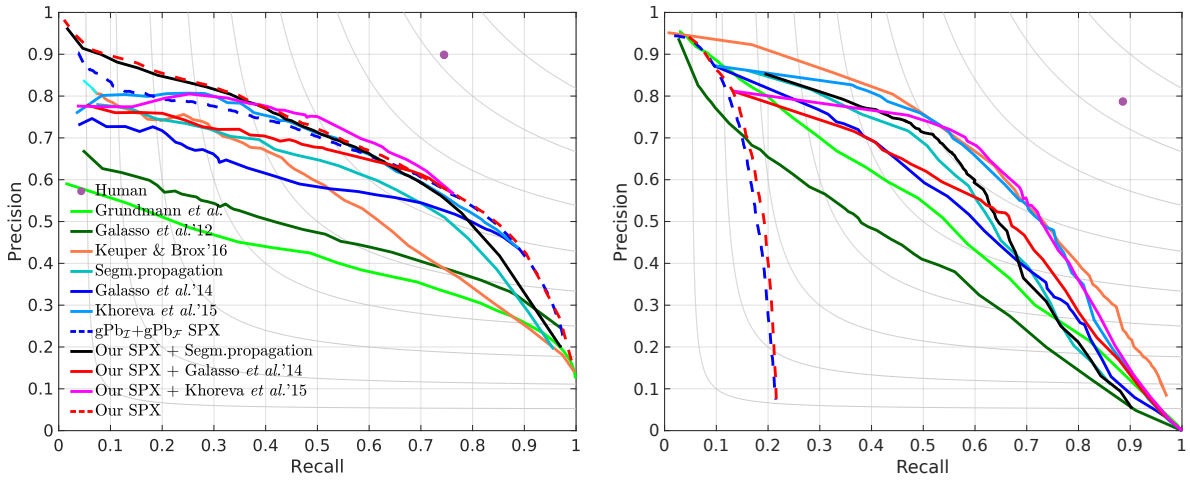
improving temporal smoothness of the boundaries and having better accuracy on a frame by frame basis. Thus we aim for the smallest negative impact on BPR. In our preliminary experiments the key for temporal smoothing was to use the right merging strategy (Section 8.6.1.4). We expect temporal smoothing to improve temporal consistency.

Object proposals. Adding OP$(\text{SE}_\mathcal{F})$ over OP$(\text{SE}_\mathcal{I})$ also improves the BPR measure (see OP$(\text{SE}_\mathcal{F})$ +OP$(\text{SE}_\mathcal{I})$ =β in Figure 8.4), particularly in the high-precision area. Merging it with other cues helps to push BPR for our final frame-by-frame result.

Combination and re-training. Combining all cues together improves the BPR metric with respect to only using appearance cues, we reach 59% AP versus 52% with appearance only (see Section 8.6.1.5). This results are better than the gPb$_\mathcal{I}$+gPb$_\mathcal{F}$ baseline (51% AP, used in Galasso *et al.* (2014)). Similar to the appearance-only case, re-training over VSB100 gives an important boost (70% AP). In this case not only $\text{SE}_\mathcal{I}$ is re-trained but also $\text{SE}_\mathcal{F}$ (over EpicFlow).

Figure 8.2 compares superpixels extracted from the proposed method (α+β model without re-training for fair comparison) with other methods. Our method reaches top results on both boundary precision and recall. Unless otherwise specified, all following "Our SPX" results correspond to superpixels generated from the hierarchical image segmentation (Arbeláez *et al.*, 2011) based on the proposed boundary estimation α+β re-trained on VSB100.

Conclusion. Temporal cues are effective at improving the boundary detection for video sequences. Because we use multiple ingredients based on machine learning, training on VSB100 significantly improves quality of boundary estimates on a per-frame basis (BPR).



(a) BPR on test set

(b) VPR on test set

Figure 8.6: Comparison of video segmentation algorithms with/without our improved superpixels, on the test set of VSB100 (Galasso *et al.*, 2013). Dashed lines indicate only frame-by-frame processing. See Table 8.1 and Section 8.7.2 for details.

8.7 VIDEO SEGMENTATION RESULTS

In this section we show results for the video segmentation methods of Galasso *et al.* (2013, 2014) with superpixels extracted from the proposed boundary estimation. So far we have only evaluated boundaries of frame-by-frame hierarchical segmentation. For all further experiments we will use the best performing model trained on VSB100, which uses image domain and temporal cues, proposed in Section 8.6 (we refer to $(\alpha + \beta)$ model, see Figure 8.4). Superpixels extracted from our boundaries help to improve video segmentation and generalizes across different datasets.

8.7.1 Validation set results

We use two baseline methods ((Galasso *et al.*, 2014, 2013), see Section 8.3) to show the advantage of using the proposed superpixels, although our approach is directly applicable to any graph-based video segmentation technique. The baseline methods originally employ the superpixels proposed in Arbeláez *et al.* (2011); Galasso *et al.* (2012), which use the boundary estimation gPb$_{\mathcal{I}}$+gPb$_{\mathcal{F}}$ to construct a segmentation.

For the baseline method of Galasso *et al.* (2014) we build a graph, where superpixels generated from the hierarchical image segmentation based on the proposed boundary estimation are taken as nodes. Following Galasso *et al.* (2014) we select the hierarchy level of image segmentation to extract superpixels (threshold over the ultrametric contour map) by a grid search on the validation set. We aim for the level which gives the best video segmentation performance, optimizing for both BPR and

Algorithm	BPR			VPR			Length	NCL
	ODS	OSS	AP	ODS	OSS	AP	$\mu\,(\delta)$	μ
Human	0.81	0.81	0.67	0.83	0.83	0.70	83.2(40.0)	12
Grundmann *et al.* (2010)	0.47	0.54	0.41	0.52	0.55	0.52	87.7(34.0)	19
Galasso *et al.* (2012)	0.51	0.56	0.45	0.45	0.51	0.42	80.2(37.6)	8
Yi and Pavlovic (2015)	0.63	0.67	0.60	**0.64**	**0.67**	0.65	35.8(38.9)	167
Keuper and Brox (2016)	0.56	0.63	0.56	**0.64**	0.66	**0.67**	1.1(0.7)	963
Galasso *et al.* (2013)	0.61	0.65	0.59	0.59	0.62	0.56	25.5(36.5)	258
Our SPX + Galasso *et al.* (2013)	**0.64**	**0.69**	**0.67**	**0.61**	**0.63**	**0.57**	22.2(34.4)	217
Galasso *et al.* (2014)	0.62	0.66	**0.54**	0.55	0.59	**0.55**	61.3(40.9)	80
Our SPX + Galasso *et al.* (2014)	**0.66**	**0.68**	0.51	**0.58**	**0.61**	**0.55**	70.4(40.2)	15
Khoreva *et al.* (2015)	0.64	**0.70**	**0.61**	0.63	0.66	**0.63**	83.4(35.3)	50
Our SPX + Khoreva *et al.* (2015)	**0.66**	**0.70**	0.55	**0.64**	**0.67**	0.61	79.4(35.6)	50

Table 8.1: Comparison of video segmentation algorithms with our proposed method based on the improved superpixels, on the test set of VSB100 (Galasso *et al.*, 2013). The table shows BPR and VPR and length statistics (mean μ, standard deviation δ, no. clusters NCL), see Figure 8.6 and Section 8.7.2 for details.

VPR.

Figure 8.5 presents results on the validation set of VSB100. The dashed curves indicate frame-by-frame segmentation and show (when touching the continuous curves) the chosen level of hierarchy to extract superpixels. As it appears in the plots, our superpixels help to improve video segmentation performance on BPR and VPR for both baseline methods (Galasso *et al.*, 2013, 2014). Figure 8.5c shows the performance of video segmentation with the proposed superpixels per video sequence. Our method improves most on hard cases, where the performance of the original approach was quite low, OSS less than 0.5.

8.7.2 Test set results

VSB100. Figure 8.6 and Table 8.1 show the comparison of the baseline methods (Galasso *et al.*, 2013, 2014) with and without superpixels generated from the proposed boundaries, and with other video segmentation algorithms on the test set of VSB100. For extracting per-frame superpixels from the constructed hierarchical segmentation we use the level selected on the validation set.

As shown in the plots and the table, the proposed method improves the baselines considered. The segmentation propagation (Galasso *et al.*, 2013) method improves $\sim$5 percent points on the BPR metrics, and 1$\sim$2 points on the VPR metrics. This supports that employing temporal cues helps to improve temporal consistency across frames. Our superpixels also boosts the performance of the approach from Galasso *et al.* (2014).

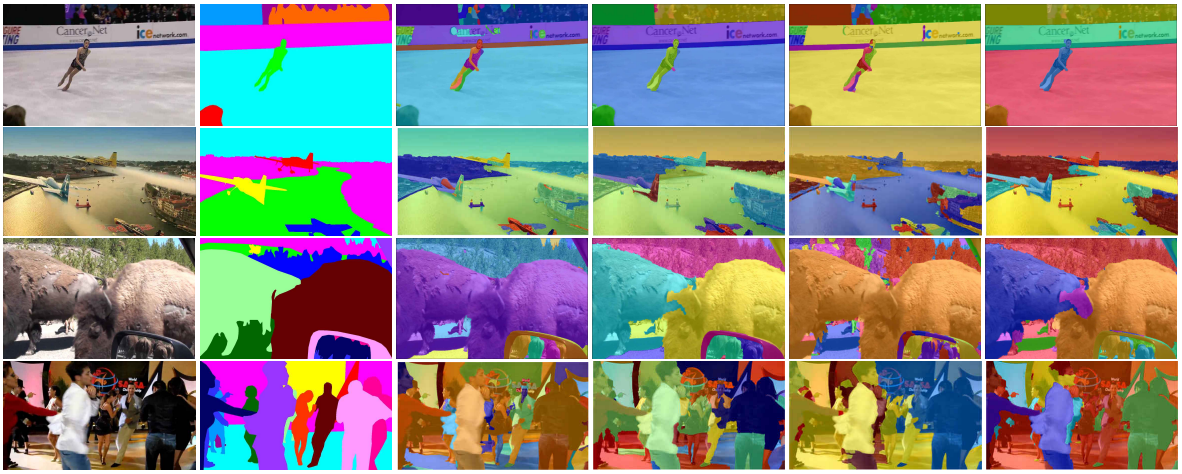


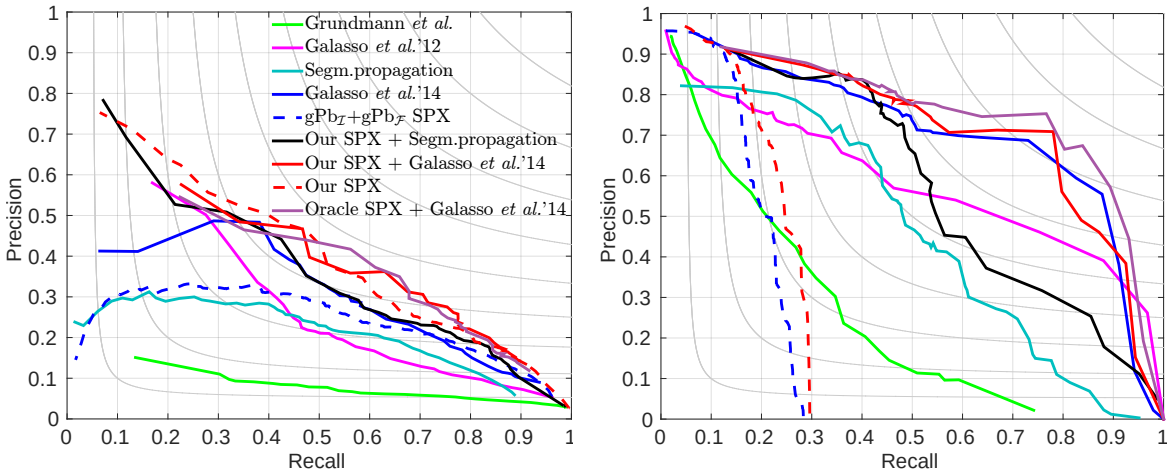
Figure 8.7: Comparison of video segmentation results of Galasso *et al.* (2013) and Galasso *et al.* (2014) with our proposed superpixels to one human ground truth. The last row shows a failure case for all methods.

Employing our method for graph-based video segmentation also benefits computational load, since it depends on the number of nodes in the graph (number of generated superpixels). On average the number of nodes is reduced by a factor of 2.6, 120 superpixels per frame versus 310 in Galasso *et al.* (2014). This leads to $\sim$ 45% reduction in runtime and memory usage for video segmentation.

Given the videos and their optical flow, the superpixel computation takes 90% of the total time and video segmentation only 10% (for both Galasso *et al.* (2014) and our SPX+Galasso *et al.* (2014)). Our superpixels are computed 20% faster than $\text{gPb}_{\mathcal{I}}$+$\text{gPb}_{\mathcal{F}}$ (the bulk of the time is spent in OP $(\cdot)$). The overall time of our approach is 20% faster than Galasso *et al.* (2014).

Qualitative results are shown in Figure 8.7. Superpixels generated from the proposed boundaries allow the baseline methods (Galasso *et al.*, 2013, 2014) to better distinguish visual objects and to limit label leakage due to inherent temporal smoothness of the boundaries. Qualitatively the proposed superpixels improve video segmentation on easy (e.g. first row of Figure 8.7) as well as hard cases (e.g. second row of Figure 8.7).

As our approach is directly applicable to any graph-based video segmentation technique we additionally evaluated our superpixels with the classifier-based graph construction method of Khoreva *et al.* (2015). The method learns the topology and edge weights of the graph using features of superpixels extracted from per-frame segmentations. We employed this approach without re-training the classifiers on the proposed superpixels. Using our superpixels allows to achieve on par performance (see Figure 8.6 and Table 8.1) while significantly reducing the runtime and memory load ($\sim$ 45%). Superpixels based on per-frame boundary estimation



(a) BPR on BMDS test set (b) VPR on BMDS test set

Figure 8.8: Comparison of video segmentation algorithms with the proposed superpixels, on BMDS (Brox and Malik, 2010). Dashed lines indicate only frame-by-frame processing (see Section 8.7.2 for details).

are also employed in Yi and Pavlovic (2015). However, we could not evaluate its performance with our superpixels as the code is not available under open source.

BMDS. Further we evaluate the proposed method on BMDS (Brox and Malik, 2010) to show the generalization of our superpixels across datasets. We use the same model trained on VSB100 for generating superpixels and the hierarchical level of boundary map as validated by a grid search on the training set of BMDS. The results are presented in Figure 8.8. Our boundaries based superpixels boost the performance of the baseline methods (Galasso *et al.*, 2013, 2014), particularly for the BPR metric (up to 4-12%).

Oracle. Additionally we set up the oracle case for the baseline of Galasso *et al.* (2014) (purple curve in Figure 8.8) by choosing the hierarchical level to extract superpixels from the boundary map for each video sequence individually based on its performance (we considered OSS measures for BPR and VPR of each video). The oracle result indicates that the used fixed hierarchical level is quite close to an ideal video-per-video selection.

8.8 CONCLUSION

The presented experiments have shown that boundary based superpixels, extracted via hierarchical image segmentation, are a better starting point for graph-based video segmentation than classical superpixels. However, the segmentation quality depends directly on the quality of the initial boundary estimates.

Over the state-of-the-art methods such as $SE_{\mathcal{I}}$ (Dollár and Zitnick, 2015), our results show that we can significantly improve boundary estimates when using cues from object proposals, globalization, and by merging with optical flow cues. When using superpixels built over these improved boundaries, we observe consistent improvement over two different video segmentation methods (Galasso *et al.*, 2013, 2014) and two different datasets (VSB100, BMDS). The results analysis indicates that we improve most in the cases where baseline methods degrade.

For future work we are encouraged by the promising results of object proposals. We believe that there is room for further improvement by integrating more semantic notions of objects into video segmentation.

Part III

LEARNING TO TRACK OBJECTS IN VIDEOS VIA CNNS

In this part of the thesis we focus on pixel-level object tracking, also referred to as semi-supervised video object segmentation. Given the first frame labelled with the object mask, the goal is to accurately segment the same instance in future frames. Recently CNNs have been proposed to solve this task. However, pixel-level tracking can be difficult to approach via convnets, since there is a lack of large body of densely, pixel-wise annotated video data. In this part we address this limitation and show that fully annotated video data is not necessary to achieve high-quality video segmentation results. We propose to use static images instead to train the network in Chapter 9 and to generate in-domain synthetic data in Chapter 10.

In Chapter 9 we treat the problem as guided instance segmentation and utilize a semantic labelling convnet for frame-by-frame segmentation. The temporal context is enabled by using the guidance from the estimated mask of the previous frame as an additional input channel to the network. We additionally fine-tune the model per-video using the first frame annotation to make the output sensitive to the specific object being tracked. The proposed framework is extended in Chapter 10 by efficiently integrating motion cues along with the appearance via a two-stream mask refinement network and by an elaborate data augmentation scheme, which creates a large number of training examples from the first annotated frame and reduces the dependence on large video and image datasets for training.

9 LEARNING VIDEO OBJECT SEGMENTATION FROM STATIC IMAGES

Inspired by recent advances of deep learning in object segmentation and tracking, in this chapter we introduce the concept of convnet-based guidance applied to video object segmentation. Our model proceeds on a per-frame basis, guided by the output of the previous frame towards the object of interest in the next frame. We demonstrate that highly accurate object segmentation in videos can be enabled by using a convnet trained with static images only. The key component of our approach is a combination of offline and online learning strategies, where the former produces a refined mask from the previous' frame estimate and the latter allows to capture the appearance of the specific object instance.

Our method can handle different types of input annotations such as bounding boxes and segments while leveraging an arbitrary amount of annotated frames. Therefore the proposed system is suitable for diverse applications with different requirements in terms of accuracy and efficiency.

9.1 INTRODUCTION

Convolutional neural networks have shown outstanding performance in many fundamental areas in computer vision, enabled by the availability of large-scale annotated datasets (e.g., ImageNet classification (Krizhevsky *et al.*, 2012; Russakovsky *et al.*, 2015)). However, some important challenges in video processing can be difficult to approach using convnets, since creating a sufficiently large body of densely, pixel-wise annotated video data for training is usually prohibitive.

One example of such domain is video object segmentation. Given only one or a few frames annotated with segmentation masks of a particular object instance, the task of video object segmentation is to accurately segment the same instance in all other frames of the video. Current top performing approaches either interleave box tracking and segmentation (Xiao and Lee, 2016), or propagate the first frame mask annotation in space-time via CRF or GrabCut-like techniques (Tsai *et al.*, 2016; Maerki *et al.*, 2016).

One of the key insights and contributions of this work is that fully annotated video data is not necessary. We demonstrate that highly accurate video object segmentation can be enabled using a convnet trained with *static images* only. We show that a convnet designed for semantic image segmentation (Chen *et al.*, 2016b) can be utilized to perform per-frame instance segmentation, i.e., segmentation of generic objects while distinguishing different instances of the same class. For each new video frame the network is guided towards the object of interest by feeding

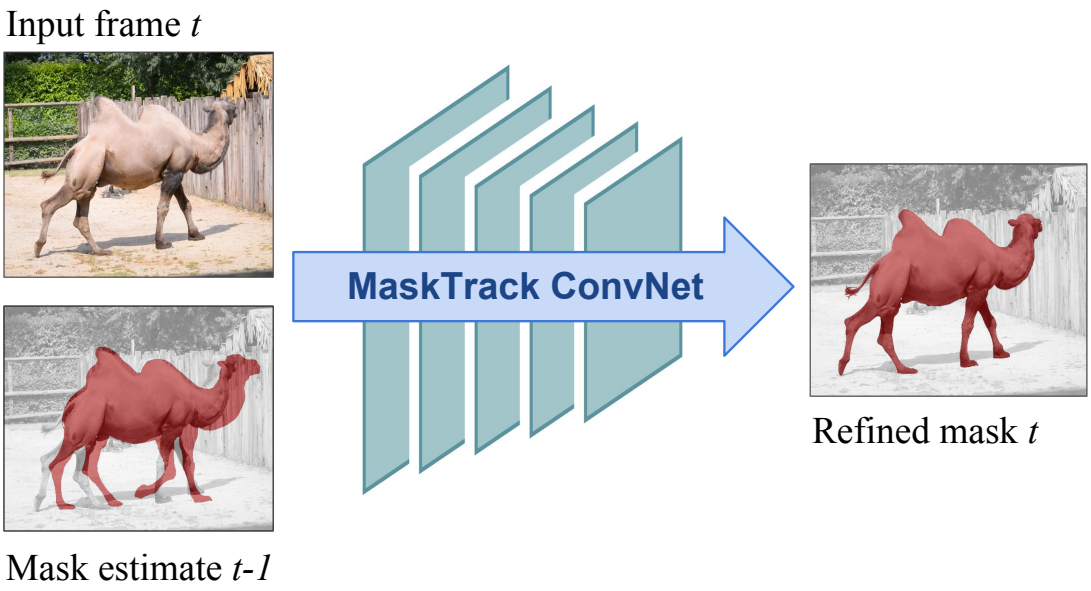


Figure 9.1: Given a rough mask estimate from the previous frame $t-1$ we train a convnet to provide a refined mask output for the current frame t.

in the previous' frame mask estimate. We therefore refer to our approach as *guided instance segmentation*.

Our system is efficient due to its feed-forward architecture and can generate high quality results in a single pass over the video, without the need for considering more than one frame at a time. This is in stark contrast to many other video segmentation approaches, which usually require global connections over multiple frames or even the whole video sequence in order to achieve coherent results. Futhermore, our method can handle different types of annotations and even simple bounding boxes as input are sufficient to obtain competitive results, making our method flexible with respect to various practical applications with different requirements in terms of human supervision.

Key to the video segmentation quality of our approach is the combination of offline and online learning strategies. In the offline phase, we use deformation and coarsening on the image masks in order to train the network to produce accurate output masks from their rough estimates. An online training phase extends ideas from previous works on object tracking (Danelljan *et al.*, 2016; Nam and Han, 2016) to the task of video segmentation and enables the method to be easily optimized with respect to an object of interest in a novel input video.

The result is a single, homogeneous system that compares favorably to most classical approaches on three extremely heterogeneous video segmentation benchmarks, despite using the same model and parameters across all videos. We provide a detailed ablation study and explore the impact of varying number and types of annotations. Moreover, we discuss extensions of the proposed model, allowing to improve the quality even further.

9.2 METHOD

We approach the video object segmentation problem from a different perspective, which we call convnet-based *guided instance segmentation*. For each new frame we wish to label pixels as object/non-object of interest, for this we build upon the architecture of the existing pixel labelling convnet and train it to generate per-frame instance segments. We pick DeepLabv2 (Chen *et al.*, 2016b), but our approach is agnostic of the specific architecture selected. The challenge is then: how to inform the network which instance to segment? We solve this by using two complementary strategies. First we guide the network towards the instance of interest by feeding in the previous' frame mask estimate during offline training (Section 9.2.1). Second, we employ online training to fine-tune the model to incorporate specific knowledge of the object instance (Section 9.2.2).

9.2.1 Offline training

In order to guide the pixel labeling network to segment the object of interest, we begin by expanding the convnet input from RGB to RGB+mask channels. The extra mask channel is meant to provide an estimate of the visible area of the object in the current frame, its approximate location and shape. We can then train the labelling convnet to output an accurate segmentation of the object, given as input the current image and a rough estimate of the object mask. Our tracking network is de-facto a "mask refinement" network.

There are two key observations that make this approach practical. First, very rough input masks are enough for our trained network to provide sensible output segments. Even a large bounding box as input will result in a reasonable output (see Section 9.4.2). The main role of the input mask is to point the convnet towards the correct object instance to segment. Second, this particular approach does not require us to use video as training data, such as done in Caelles *et al.* (2017b); Held *et al.* (2016); Bertinetto *et al.* (2016); Nam and Han (2016). Because we only use a mask as additional input, instead of an image crop as in Held *et al.* (2016); Bertinetto *et al.* (2016), we can synthesize training samples from single frame instance segmentation annotations. This allows us to train from a large set of diverse images, instead of having to rely on scarce video annotations.

Figure 9.1 shows our simplified model. To simulate the noise of the previous frame output, during offline training, we generate input masks by deforming the annotations via affine transformation as well as non-rigid deformations via thin-plate splines (Bookstein, 1989), followed by a coarsening step (dilation morphological operation) to remove details of the object contour. We apply this data generation procedure over a dataset of $\sim 10^4$ images containing diverse object instances. At test time, given the mask estimate at time $t-1$, we apply the dilation operation and use the resulting rough mask as input for object segmentation in frame t.

The affine transformations and non-rigid deformations aim at modelling the

(a) Annotated image (b) Example training masks

Figure 9.2: Examples of training masks. From one annotated image multiple masks are generated. The generated masks mimic plausible object shapes on the preceding frame.

expected motion of an object between two frames. The coarsening permits us to generate training samples that resembles the test time data, simulating the blobby shape of the output mask given from the previous frame by the convnet. These two ingredients make the estimation more robust to noisy segmentation estimates while helping to avoid accumulation of errors from the preceding frames. The trained convnet has learnt to do guided instance segmentation similar to networks like SharpMask (Pinheiro *et al.*, 2016), DeepMask (Pinheiro *et al.*, 2015) and Hypercolumns (Hariharan *et al.*, 2015), but instead of taking a bounding box as guidance, we can use an arbitrary input mask. The training details are described in Section 9.3.

When using offline training only, the segmentation procedure consists of two steps: the previous frame mask is coarsened and then fed into the trained network to estimate the current frame mask. Since objects have a tendency to move smoothly through space, the object mask in the preceding frame will provide a good guess in the current frame and simply copying the coarse mask from the previous frame is enough. This approach is fast and already provides good results. We also experimented using optical flow to propagate the mask from one frame to the next, but found the optical flow errors to offset the gains.

With only the offline trained network, the proposed approach allows us to achieve competitive performance compared to previously reported results (see Section 9.4.2). However, the performance can be further improved by integrating online training strategy as described in the next section.

9.2.2 Online training

For further boosting the video segmentation quality, we borrow and extend ideas that were originally proposed for object tracking. Current top performing tracking techniques (Danelljan *et al.*, 2016; Nam and Han, 2016) use some form of online training. We thus consider improving results by adding online fine-tuning as a second strategy.

The idea is to use, at test time, the segment annotation of the first video frame as additional training data. Using augmented versions of this single frame annotation,

we proceed to fine-tune the model to become more specialized for the specific object instance at hand. We use a similar data augmentation as for offline training. On top of affine and non-rigid deformations for the input mask, we also add image flipping and rotations. We generate $\sim 10^3$ training samples from this single annotation, and proceed to fine-tune the model previously trained offline.

With online fine-tuning, the network weights partially capture the appearance of the specific object being tracked. The model aims to strike a balance between general instance segmentation (so as to generalize to the object changes), and specific instance segmentation (so as to leverage the common appearance across video frames). The details of the online fine-tuning are provided in Section 9.3. In our experiments we only perform fine-tuning using the annotated frame(s).

To the best of our knowledge our approach is the first to use a pixel labelling network (like DeepLabv2 (Chen *et al.*, 2016b)) for the task of video object segmentation. We name our full approach, using both offline and online training, `MaskTrack`.

9.2.3 Variants

Additionally we consider variations of the proposed model. First, we demonstrate that our approach is flexible and could handle different types of input annotations, using less supervision in the first frame annotation. Second, we describe how motion information could be easily integrated in the system, improving the quality of the object segments.

Box annotation. In this paragraph, we discuss a variant named $\texttt{MaskTrack}_{Box}$, that takes a bounding box annotation in the first frame as an input supervision instead of a segmentation mask. To this end, we train a similar convnet that fed with a bounding-box annotation as an input outputs a segment. Once the first frame bounding box is converted to a segment, we switch back to the `MaskTrack` model that uses as guidance the output mask from the previous frame.

Optical flow. On top of `MaskTrack`, we consider employing optical flow as a source of additional information to guide the segmentation. Given a video sequence, we compute the optical flow using EpicFlow (Revaud *et al.*, 2015) with Flow Fields matches (Bailer *et al.*, 2015) and convolutional boundaries (Maninis *et al.*, 2017). In parallel to the vanilla `MaskTrack`, we proceed to compute a second output mask using the magnitude of the optical flow as input image (replicated into a three channel image). The model is used as-is, without retraining. Although it has been trained on RGB images, this strategy works as object flow magnitude roughly looks like a gray-scale object, and still captures useful object shape information, see examples in Figure 9.3. Using the RGB model allows to avoid training the convnet on video datasets annotated with masks. We then fuse by averaging the output scores given by the two parallel networks, respectively fed with RGB images and optical flow magnitude as input. As shown in Table 9.1, optical flow provides complementary information to `MaskTrack` with RGB images, improving the overall performance.

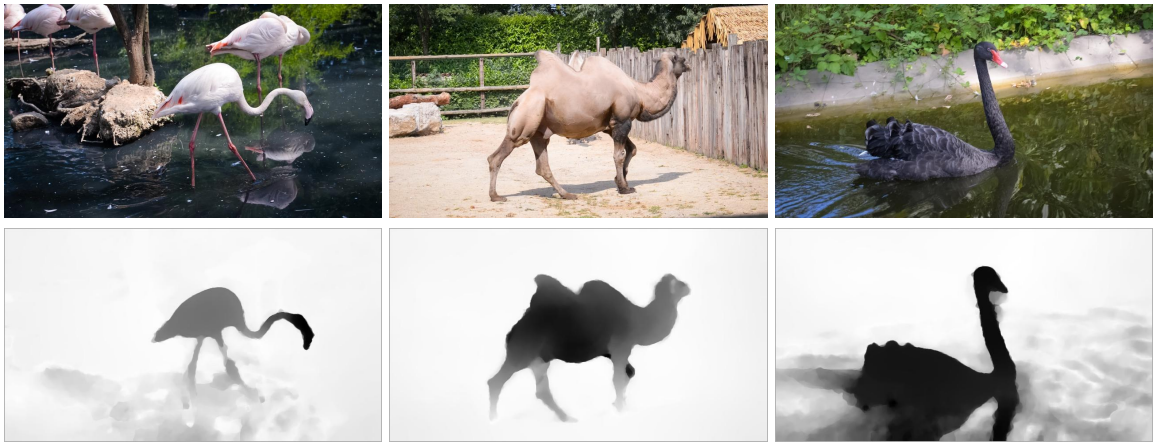


Figure 9.3: Examples of optical flow magnitude images. Top: RGB images. Bottom: corresponding motion magnitude estimates encoded into as gray-scale images.

9.3 NETWORK IMPLEMENTATION AND TRAINING

Following, we describe the implementation details of our approach. Specifically, we provide additional information regarding the network initialization, the offline and online training strategies and the data augmentation.

Network. For all our experiments we use the training and test parameters of DeepLabv2-VGG network (Chen *et al.*, 2016b). The model is initialized from a VGG16 network pre-trained on ImageNet (Simonyan and Zisserman, 2015). For the extra mask channel of filters in the first convolutional layer we use gaussian initialization. We also tried zero initialization, but observed no difference.

Offline training. The advantage of our method is that it does not require expensive pixel-wise video annotations for training. Thus we can employ existing image datasets. However, in order for our model to generalize well across different videos, we avoid training on datasets that are biased towards certain semantic classes, such as COCO (Lin *et al.*, 2014) or Pascal (Everingham *et al.*, 2015). Instead we combine images and annotations from several saliency segmentation datasets (ECSSD (Shi *et al.*, 2016), MSRA10K (Cheng *et al.*, 2015b), SOD (Movahedi and Elder, 2010), and PASCAL-S (Li *et al.*, 2014)), resulting in an aggregated set of 11 282 training images.

The input masks for the extra channel are generated by deforming the binary segmentation masks via affine transformation and non-rigid deformations, as discussed in Section 9.2.1. For affine transformation we consider random scaling ($\pm$5% of object size) and translation ($\pm$10% shift). Non-rigid deformations are done via thin-plate splines (Bookstein, 1989) using 5 control points and randomly shifting the points in x and y directions within $\pm$10% margin of the original segmentation mask width and height. Next, the mask is coarsened using dilation operation with 5 pixel radius. This mask deformation procedure is applied over all object instances in the

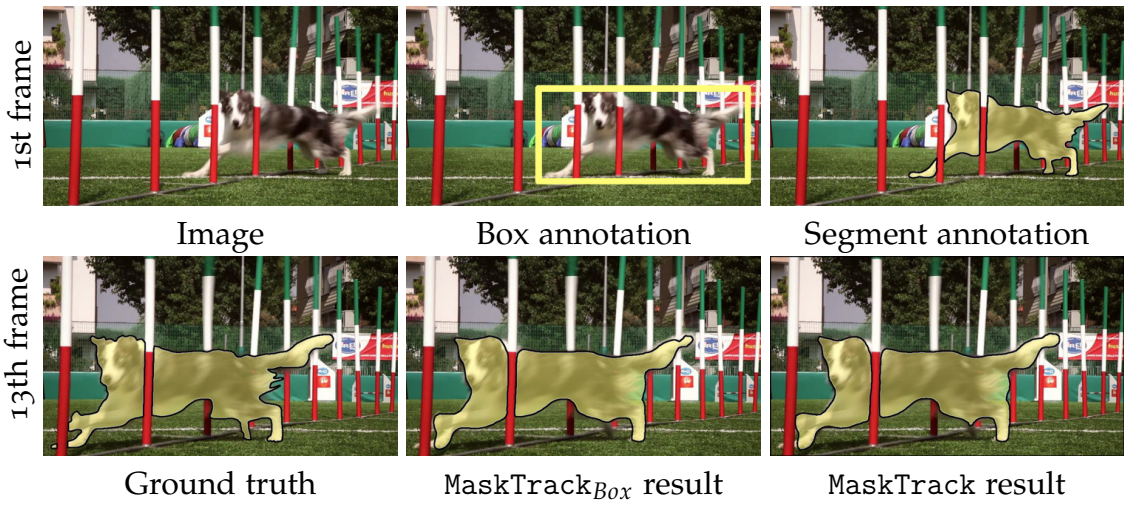


Figure 9.4: By propagating annotation from the 1st frame, either from segment or just bounding box annotations, our system generates results comparable to ground truth.

training set. For each image two different masks are generated.

The convnet training parameters are identical to those proposed in Chen *et al.* (2016b). Therefore we use stochastic gradient descent (SGD) with mini-batches of 10 images and a polynomial learning policy with initial learning rate of 0.001. The momentum and weight decay are set to 0.9 and 0.0005, respectively. The network is trained for 20k iterations.

Online training. For online adaptation we fine-tune the model previously trained offline on the first frame for 200 iterations with training samples generated from the first frame annotation. We augment the first frame by image flipping and rotations as well as by deforming the annotated masks for an extra channel via affine and non-rigid deformations with the same parameters as for the offline training. This results in an augmented set of $\sim 10^3$ training images. The network is trained with the same learning parameters as for offline training, fine-tuning all convolutional layers.

At test time our base `MaskTrack` system runs at about 12 seconds per frame (averaged over DAVIS, amortizing the online fine-tuning time over all video frames), which is a magnitude faster compared to `ObjFlow` (Tsai *et al.*, 2016) (takes 2 minutes per frame, averaged over DAVIS).

9.4 RESULTS

In this section we describe our evaluation protocol (Section 9.4.1), study the importance of the different components of our system (Section 9.4.2), and report results comparing to state-of-the-art techniques over three datasets (Section 9.4.3),

as well as comparing the effects of different amounts of annotation on the resulting segmentation quality (Section 9.4.4).

9.4.1 Experimental setup

Datasets. We evaluate the proposed approach on three different video object segmentation datasets: DAVIS (Perazzi *et al.*, 2016), YoutubeObjects (Prest *et al.*, 2012), and SegTrack-v2 (Li *et al.*, 2013). These datasets include assorted challenges such as appearance change, occlusion, motion blur and shape deformation.

DAVIS (Perazzi *et al.*, 2016) consists of 50 high quality videos, totaling 3 455 frames. Pixel-level segmentation annotations are provided for each frame, where one single object or two connected objects are separated from the background.

YoutubeObjects (Prest *et al.*, 2012) includes videos with 10 object categories. We consider the subset of 126 videos with more than 20 000 frames, for which the pixel-level ground truth segmentation masks are provided by Jain and Grauman (2014).

SegTrack-v2 (Li *et al.*, 2013) contains 14 video sequences with 24 objects and 947 frames. Every frame is annotated with a pixel-level object mask. As instance-level annotations are provided for sequences with multiple objects, each specific instance segmentation is treated as separate problem.

Evaluation. We evaluate using the standard mIoU metric: intersection-over-union of the estimated segmentation and the ground truth binary mask, also known as *Jaccard Index*, averaged across videos. For DAVIS we use the provided benchmark code (Perazzi *et al.*, 2016), which excludes the first and the last frames from the evaluation. For YoutubeObjects and SegTrack-v2 only the first frame is excluded.

Previous works used different evaluation procedures. To ensure a consistent comparison between methods, when needed, we re-computed scores from the publicly available output masks, or reproduced the results using the available open source code. In particular, we collected new results for `ObjFlow` (Tsai *et al.*, 2016) and `BVS` (Maerki *et al.*, 2016) in order to present other methods with results across the three datasets.

9.4.2 Ablation study

We first study different ingredients of our method. We experiment on the DAVIS dataset and measure the performance using the mean intersection-over-union metric (mIoU). Table 9.1 shows the importance of each of the ingredients described in Section 9.2 and reports the improvement of adding extra components to the `MaskTrack` model.

Add-ons. We first study the effect of adding a couple of ingredients on top of our base `MaskTrack` system, which are specifically fine-tuned for DAVIS. We see that optical flow provides complementary information to the appearance, boosting

Aspect	System variant	mIoU	ΔmIoU
Add-ons	MaskTrack+Flow+CRF	**80.3**	+1.9
	MaskTrack+Flow	78.4	+3.6
	MaskTrack	*74.8*	-
Training	No online fine-tuning	69.9	−4.9
	No offline training	57.6	−17.2
	Reduced offline training	73.2	−1.6
	Training on video	72.0	−2.8
Mask deformation	No dilation	72.4	−2.4
	No deformation	17.1	−57.7
	No non-rigid deformation	73.3	−1.5
Input channel	Boxes	69.6	−5.2
	No input	72.5	−2.3

Table 9.1: Ablation study of our MaskTrack method on DAVIS. Given our full system, we change one ingredient at a time, to see each individual contribution. See Section 9.4.2.

further the results (74.8 → 78.4). Adding on top a well-tuned post-processing CRF (Krähenbühl and Koltun, 2011) can gain a couple of mIoU points, reaching 80.3% mIoU on DAVIS, the best known result on this dataset.

Although optical flow can provide interesting gains, we found it to be brittle when going across different datasets. Different strategies to handle optical flow provide 1∼4% on each dataset, but none provide consistent gains across all datasets; mainly due to failure modes of the optical flow algorithms. For the sake of presenting a single model with fix parameters across all datasets, we refrain from using a per-dataset tuned optical flow in the results of Section 9.4.3.

Training. We next study the effect of offline/online training of the network. By disabling online fine-tuning, and only relying on offline training we see a ∼5 IoU percent points drop, showing that online fine-tuning indeed expands the tracking capabilities. If instead we skip offline training and only rely on online fine-tuning performance drops drastically, albeit the absolute quality (57.6 mIoU) is surprisingly high for a system trained on ImageNet+single frame.

By reducing the amount of training data from 11k to 5k we only see a minor decrease in mIoU; this indicates that even with the small amount of training data we can achieve reasonable performance. That being said, further increase of the training data volume would lead to improved results.

Additionally, we explore the effect of the offline training on video data instead of using static images. We train the model on the annotated frames of two combined datasets, SegTrack-v2 and YoutubeObjects. By switching to train on video data we observe a minor decrease in mIoU; this could be explained by lack of diversity in

the video training data due to the small scale of the existing datasets, as well as the effect of the domain shift between different benchmarks. This shows that employing static images in our approach does not result in any performance drop.

Mask deformation. We also study the influence of mask deformations. We see that coarsening the mask via dilation provides a small gain, as well as adding non-rigid deformations. All-and-all, Table 9.1 shows that the main factor affecting the quality is using any form of mask deformations when creating the training samples (both for offline and online training). This ingredient is critical for our overall approach, making the segmentation estimation more robust at test time to the noise in the input mask.

Input channel. Next we experiment with different variants of the extra channel input. Even by changing the input from segments to boxes, a model trained for this modality still provides reasonable results. A failure mode of this approach is the generation of small blobs outside of the object. As a result the guidance from the previous frame produces a noisy box, yielding inaccurate segmentation. The error accumulates forward over the entire video sequence.

We also evaluated a model that does not use any mask input. Without the additional input channel, this pixel labelling convnet was trained offline as a salient object segmenter and fine-tuned online to capture the appearance of the object of interest. This model obtains competitive results (72.5 mIoU) on DAVIS, since the object to segment is also salient for this dataset. However, while experimenting on SegTrack-v2 and YoutubeObjects, we observed a significant drop in performance without using guidance from the previous frame mask as these two datasets have a weaker bias towards salient objects compared to DAVIS.

9.4.3 Single frame annotations

Table 9.2 presents results when the first frame is annotated with an object segmentation mask. This is the protocol commonly used on DAVIS, SegTrack-v2, and YoutubeObjects. `MaskTrack` obtains competitive performance across all three datasets. This is achieved using our purely frame-by-frame feed-forward system, using the exact same model and parameters across all datasets. Our `MaskTrack` results are obtained in a single pass, do not use any global optimization, not even optical flow. We believe this shows the promise of formulating video object segmentation from the instance segmentation perspective.

On SegTrack-v2, `JOTS` (Wen *et al.*, 2015) reported higher numbers (71.3 mIoU), however, they report tuning their method parameters' per video, and thus it is not comparable to our setup with fix-parameters. Table 9.2 also reports results for the $\texttt{MaskTrack}_{Box}$ variant described in Section 9.2.3. Starting only from box annotations on the first frame, our system still generates comparably good results (see Figure 9.4), remaining on the top three best results in all the datasets covered.

By adding additional ingredients specifically tuned for different datasets, such

Method	Dataset, mIoU		
	DAVIS	YoutbObjs	SegTrack-v2
Box oracle	45.1	55.3	56.1
Grabcut oracle	67.3	67.6	74.2
ObjFlow (Tsai *et al.*, 2016)	71.4	70.1	67.5
BVS (Maerki *et al.*, 2016)	66.5	59.7	58.4
NLC (Faktor and Irani, 2014)	64.1	-	-
FCP (Perazzi *et al.*, 2015)	63.1	-	-
W16 (Wang *et al.*, 2016)	-	59.2	-
Z15 (Zhang *et al.*, 2015b)	-	52.6	-
TRS (Xiao and Lee, 2016)	-	-	**69.1**
MaskTrack	**74.8**	**71.7**	67.4
MaskTrack_{Box}	73.7	69.3	62.4

Table 9.2: Video object segmentation results on three datasets. Compared to related state of the art, our approach provides consistently good results. On DAVIS the extended version of our system MaskTrack+Flow+CRF reaches 80.3 mIoU. See Section 9.4.3 for details.

as optical flow (see Section 9.2.3) and CRF post-processing, we can push the results even further, reaching 80.3 mIoU on DAVIS, 72.6 on YoutubeObjects and 70.3 on SegTrack-v2. Figure 9.5 presents qualitative results of the proposed MaskTrack model across three different datasets.

Attribute-based analysis. Figure 9.6 presents a more detailed evaluation on DAVIS (Perazzi *et al.*, 2016) using video attributes. The attribute based analysis shows that our generic model, MaskTrack, is robust to various video challenges presented in DAVIS. It compares favorably on any subset of videos sharing the same attribute, except camera-shake, where ObjFlow (Tsai *et al.*, 2016) marginally outperforms our approach. We observe that MaskTrack handles fast-motion and motion-blur well, which are typical failure cases for methods relying on spatio-temporal connections (Maerki *et al.*, 2016; Tsai *et al.*, 2016).

Due to the online fine-tuning on the first frame annotation of a new video, our system is able to capture the appearance of the specific object of interest. This allows it to better recover from occlusions, out-of-view scenarios and appearance changes, which usually affect methods that strongly rely on propagating segmentations on a per-frame basis.

Incorporating optical flow information into MaskTrack substantially increases robustness on all categories. As one could expect, MaskTrack+Flow+CRF better discriminates cases involving color ambiguity and salient motion. We also observe improvements in cases with scale-variation and low-resolution objects.

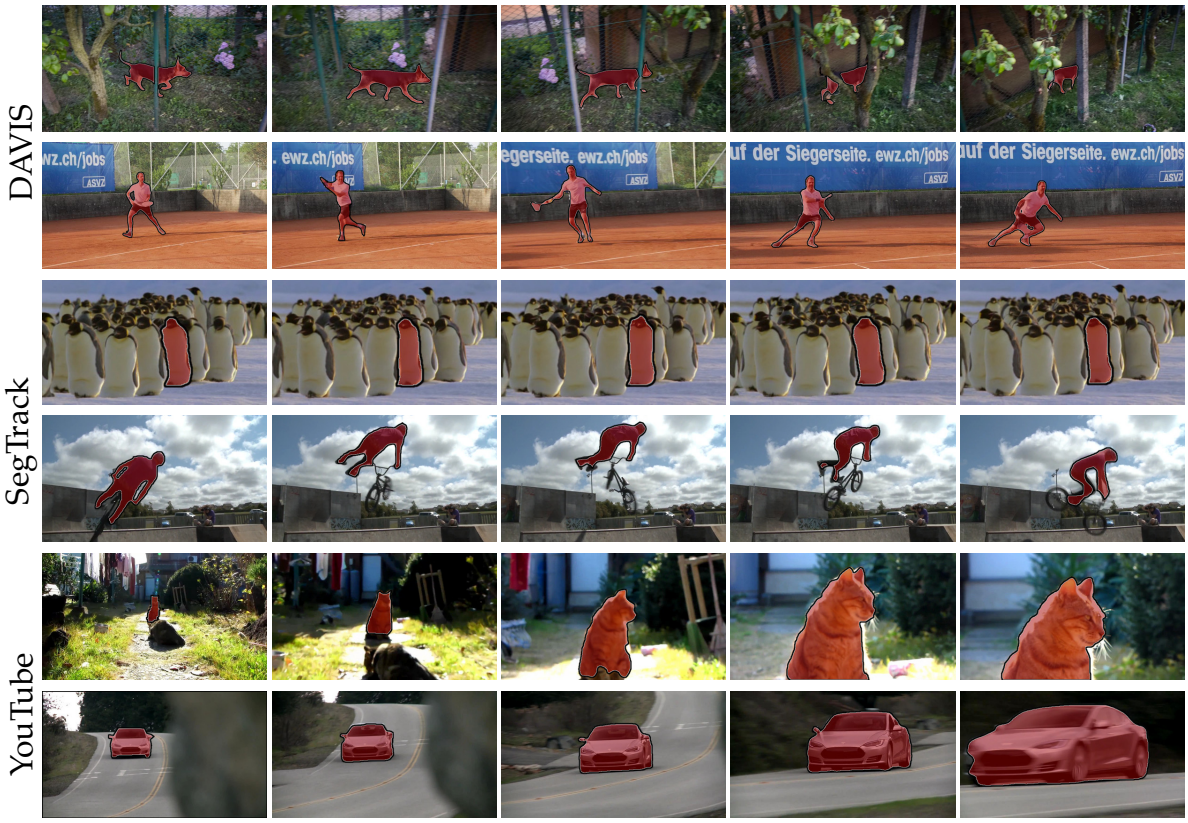


Figure 9.5: Qualitative results of three different datasets. Our algorithm is robust to challenging situations such as occlusions, fast motion, multiple instances of the same semantic class, object shape deformation, camera view change and motion blur.

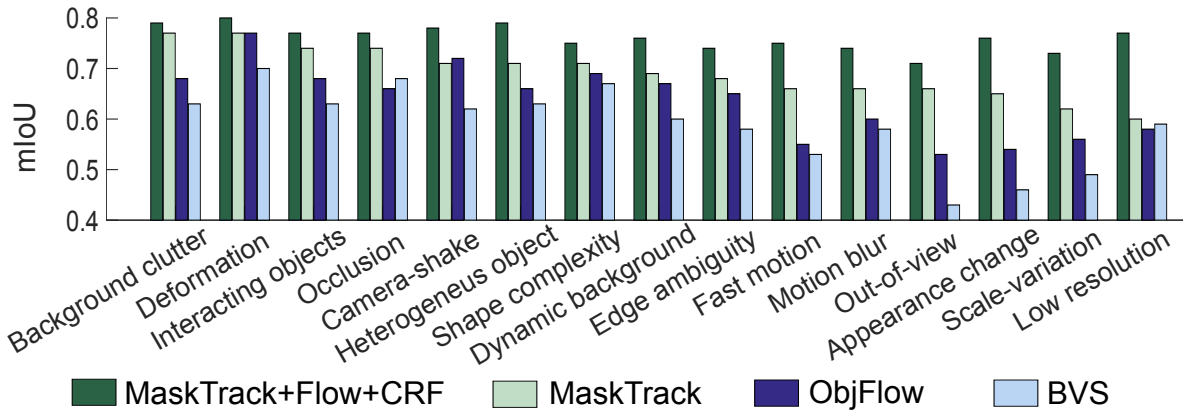


Figure 9.6: Attribute based evaluation on DAVIS.

Conclusion. With our simple, generic system for video object segmentation we are able to achieve competitive results with existing techniques, on three different datasets. These results are obtained with fixed parameters, from a forward-pass only, using only static images during offline training. We also reach good results even when using only a box annotation as starting point.

9.4.4 Multiple frame annotations

In some applications, e.g. video editing for movie production, one may want to consider more than a single frame annotation on videos. Figure 9.8 shows the segmentation quality result when considering different number of annotated frames on the DAVIS dataset. We evaluate both pixel accurate segmentation and bounding box annotations.

For these experiments, we run our method twice, forward and backwards; and for each frame we pick the result closest in time to the annotated frame (either from forward or backwards propagation). Here, the online fine-tuning uses all annotated frames instead of only the first one. For the experiments with box annotations in Figure 9.8, we use a similar procedure to $\texttt{MaskTrack}_{Box}$. Box annotations are first converted to segments, and then apply `MaskTrack` as-is, treating these as the original segment annotations. The evaluation reports the mean IoU results when annotating one frame only (same as Table 9.2), and every 40th, 30th, 20th, 10th, 5th, 3rd, and 2nd frame. Since DAVIS videos have length $\sim$100 frames, 1 annotated frame corresponds to $\sim$1%, otherwise annotations every 20th is 5% of annotated frames, 10th 10%, 5th 20%, etc. We follow the same evaluation protocol as in Section 9.4.3, ignoring first and last frames, and including the annotated frames in the evaluation (this is particularly relevant for the box annotation results).

Other than mean IoU we also show the quantile curves indicating the cutting line for the 5%, 10%, 20%, etc. lowest quality video frame results. This gives a hint of how much targeted additional annotations might be needed. The higher mean IoU of these quantiles are, the better. The baseline for these experiments consists in directly copying the ground truth annotations from the nearest annotated neighbour. For visual clarity, in Figure 9.8, we only include the mean value for the baseline experiment.

Analysis. We can see that results in Figures 9.8 show slightly different trends. When using segment annotations the baseline quality increases steadily until reaching IoU 1 when all frames are annotated. Our `MaskTrack` approach provides large gains with 30% of annotated frames or less. For instance when annotating 10% of the frames we reach mIoU 0.86, with the 20% quantile at 0.81 mIoU. This means with only 10% of annotated frames, 80% of all video frames will have a mean IoU above 0.8, which is good enough to be used for many applications, or can serve as initialization for a refinement process. With 10% of annotated frames the baseline only reaches 0.64 mIoU. When using box annotations the quality of the baseline and our method saturates. There is only so much information our instance segmenter

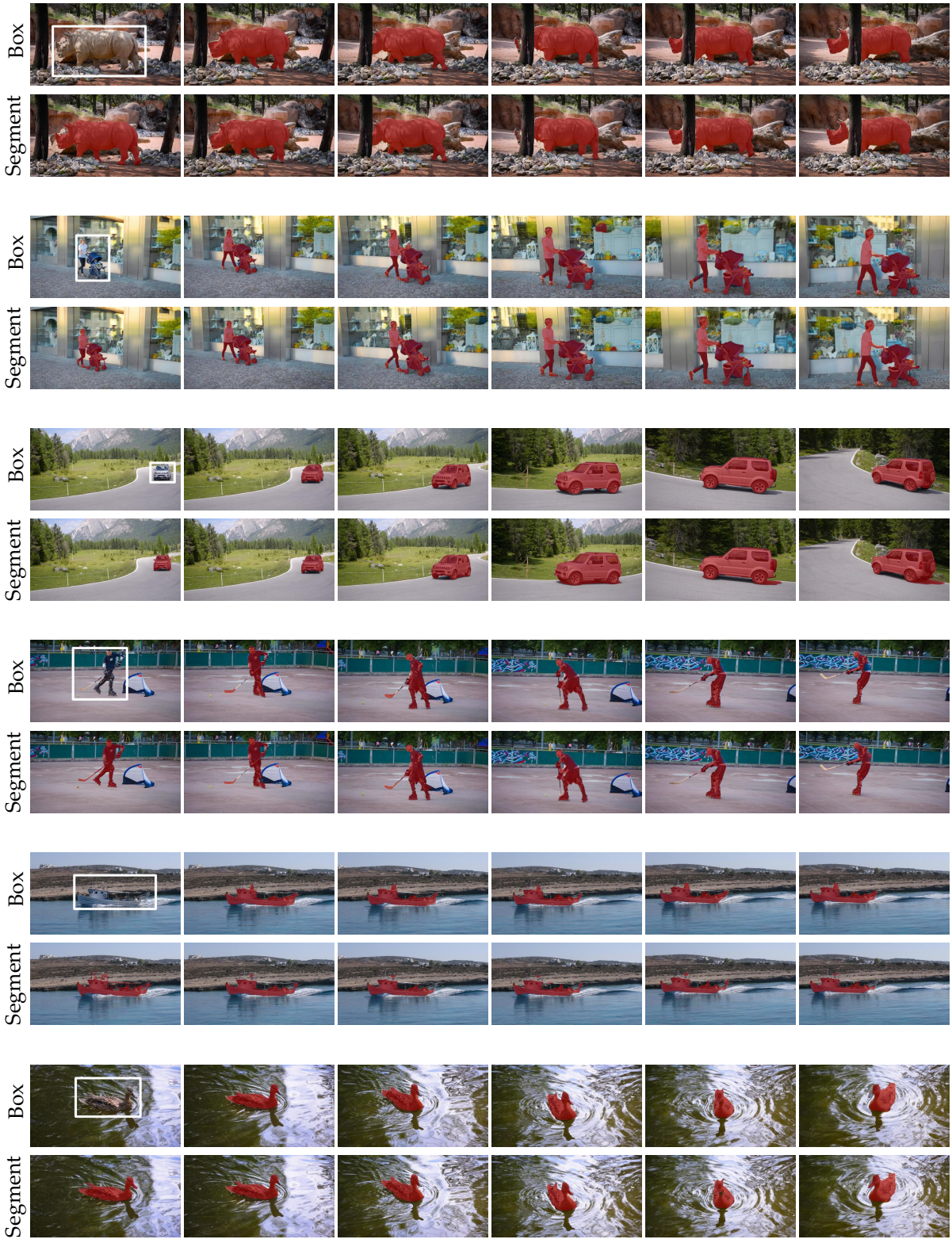
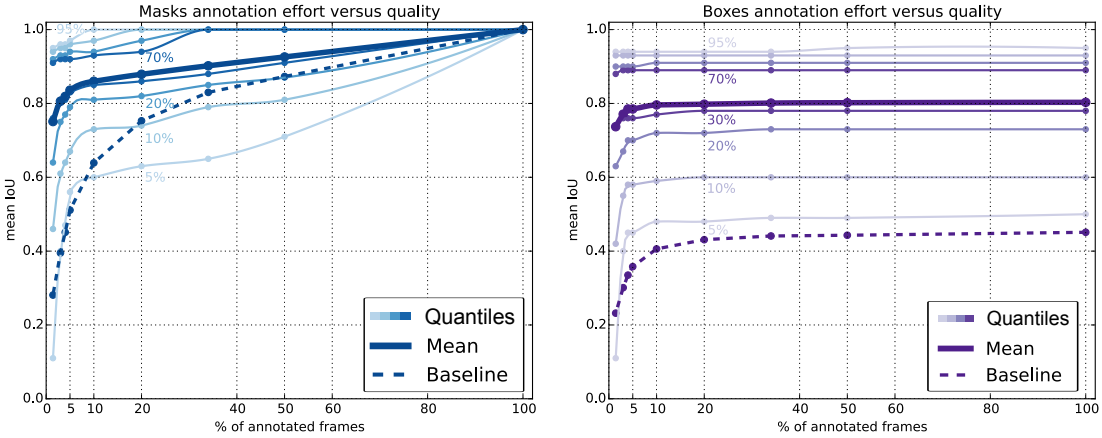


Figure 9.7: Qualitative results of MaskTrack$_{Box}$ and MaskTrack on Davis. By propagating annotation from the 1st frame, either from segment or just bounding box annotations, our system generates results comparable to ground truth.



Segment annotations. Box Annotations.

Figure 9.8: Percent of annotated frames versus video object segmentation quality. We report mean IoU, and quantiles at 5, 10, 20, 30, 70, 80, 90, and 95%. Results on DAVIS, using segment or box annotations. The baseline simply copies the annotations to adjacent frames. Discussion in Section 9.4.4.

can estimate from boxes. After 10% of annotated frames, not much additional gain is obtained. Interestingly, the mean IoU and 30% quantile here both reach $\sim$0.8 mIoU. Additionally, 70% of the frames have IoU above 0.89.

Conclusion. Results indicate that with only 10% of annotated frames we can reach satisfactory quality, even when using only bounding box annotations. We see that with moving from one annotation per video to two or three frames (1%$\rightarrow$ 3%$\rightarrow$4%) quality increases sharply, showing that our system can adequately leverage a few extra annotations per video.

9.5 CONCLUSION

We have presented a novel approach to video object segmentation. By treating video object segmentation as guided instance segmentation problem, we have proposed to use a pixel labelling convnet for frame-by-frame segmentation. By exploiting both offline and online training with image annotations only our approach is able to produce highly accurate video object segmentation. The proposed system is generic and reaches competitive performance on three extremely heterogeneous video segmentation benchmarks, using the same model and parameters across all videos. The method can handle different types of input annotations and our results are competitive even when using only bounding box annotations (instead of segmentation masks).

We provided a detailed ablation study, and explored the effect of varying the amount of annotations per video. Our results show that with only one annotation every 10th frame we can reach 85% mIoU quality. Considering we only do per-frame

instance segmentation without any form of global optimization, we deem these results encouraging to achieve high quality via additional post-processing.

We believe the use of labeling convnets for video object segmentation is a promising strategy. Future work should consider exploring more sophisticated network architectures, incorporating temporal dimension and global optimization strategies.

In Chapter 10 we extend the proposed approach with better integration of optical flow, making the gains across different datasets more stable. We also relax the dependence of using $\sim 10k$ pixel-level image annotations for training the mask tracking convnet by introducing the efficient data synthesis scheme.

LUCID DATA DREAMING FOR OBJECT TRACKING 10

Convolutional networks reach top quality in pixel-level object tracking but require large scale image or video datasets for training ($1k \sim 10k$) to deliver such results. In this chapter we propose a new training strategy which achieves state-of-the-art results across three evaluation datasets while using $20\times \sim 100\times$ less annotated data compared to the approach proposed in Chapter 9 and other competing methods. Our approach is suitable for both single and multiple object tracking.

Instead of using large training sets hoping to generalize across domains, we generate in-domain training data using the provided annotation on the first frame of each video to synthesize ("lucid dream"[4]) plausible future video frames. In-domain per-video training data allows us to train high quality appearance- and motion-based models, as well as tune the post-processing stage. This approach allows to reach competitive results even when training from only a single annotated frame, without ImageNet pre-training. Our results indicate that using a larger training set is not automatically better, and that for the tracking task a smaller training set that is closer to the target domain is more effective. This changes the mindset regarding how many training samples and general "objectness" knowledge are required for the object tracking task.

10.1 INTRODUCTION

In the last years the field of object tracking in videos has transitioned from bounding box (Kristan *et al.*, 2015, 2014, 2016) to pixel-level tracking (Li *et al.*, 2013; Prest *et al.*, 2012; Perazzi *et al.*, 2016; Vojir and Matas, 2017). Given a first frame labelled with the foreground object masks, one aims to find the corresponding object pixels in future frames. Tracking objects at the pixel level enables a finer understanding of videos and is helpful for tasks such as video editing, rotoscoping, and summarisation.

Top performing results are currently obtained using convolutional networks (convnets) (Jampani *et al.*, 2016a; Caelles *et al.*, 2017b; Perazzi *et al.*, 2017; Bertinetto *et al.*, 2016; Held *et al.*, 2016; Nam and Han, 2016). Like most deep learning techniques, convnets for pixel-level object tracking benefit from large amounts of training data. Current state-of-the-art methods rely, for instance, on pixel accurate foreground/background annotations of $\sim 2k$ video frames (Jampani *et al.*, 2016a; Caelles *et al.*, 2017b) or $\sim 10k$ images (Perazzi *et al.*, 2017). Labelling videos at the pixel level is a laborious task (compared e.g. to drawing bounding boxes for

[4] In a lucid dream the sleeper is aware that he or she is dreaming and is sometimes able to control the course of the dream.

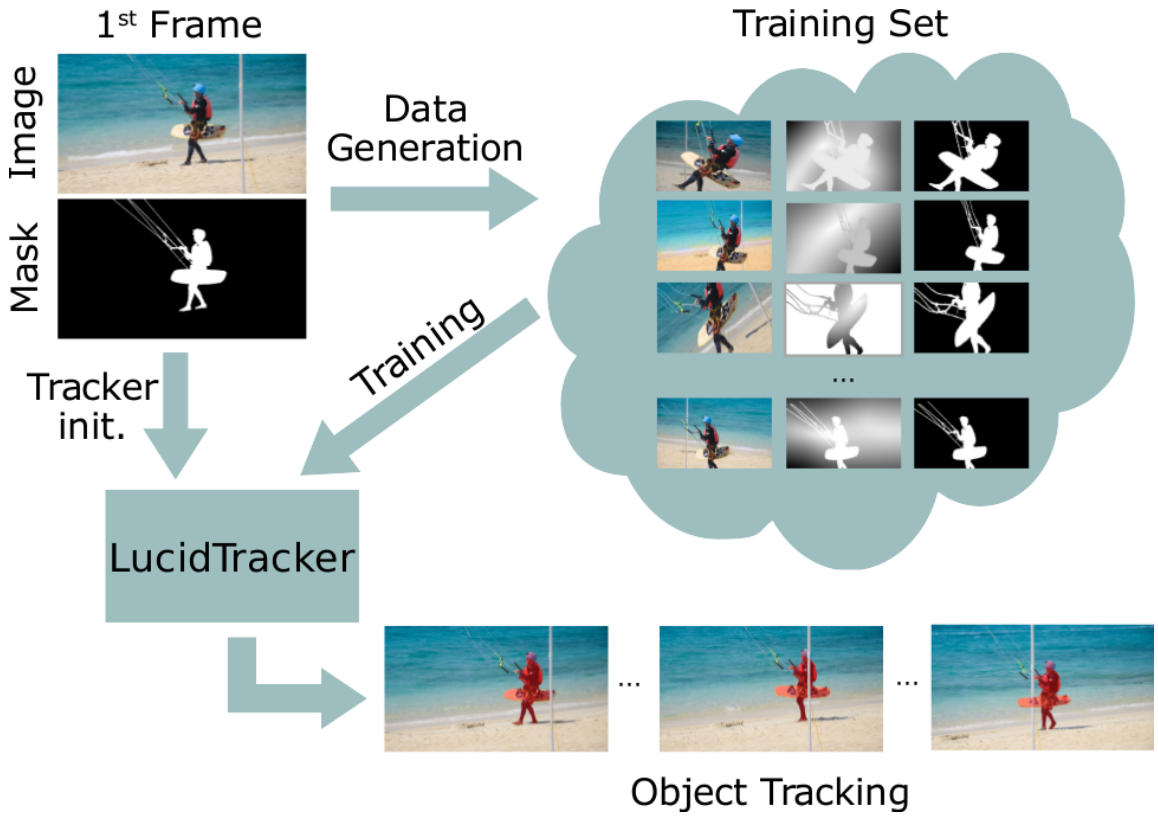


Figure 10.1: Starting from scarce annotations we synthesize in-domain data to train a specialized pixel-level object tracker for each dataset or even each video.

detection), and creating a large training set requires significant annotation effort.

In this work we aim to reduce the necessity for such large volumes of training data. It is traditionally assumed that convnets require large training sets to perform best. We show that for video object tracking having a larger training set is not automatically better and that improved results can be obtained by using $20\times\sim100\times$ less training data than previous approaches (Caelles *et al.*, 2017b; Perazzi *et al.*, 2017). The main insight of our work is that for pixel-level object tracking using few training frames ($1\sim100$) in the target domain is more useful than using large training volumes across domains ($1k\sim10k$).

To ensure a sufficient amount of training data close to the target domain, we develop a new technique for synthesizing training data particularly tailored for the object tracking scenario. We call this data generation strategy "*lucid dreaming*", where the first frame and its annotation mask are used to generate plausible future frames of the videos. The goal is to produce a large training set of reasonably realistic images which capture the expected appearance variations in future video frames, and thus is, by design, close to the target domain.

Our approach is suitable for both single and multiple object tracking. Enabled by the proposed data generation strategy and the efficient use of optical flow, we are able to achieve high quality results while using only ~100 individual annotated

training frames. Moreover, in the extreme case with only a single annotated frame (zero pre-training), we still obtain competitive tracking results.

In summary, our contributions are the following:

- We propose "lucid data dreaming", an automated approach to synthesize training data for the convnet-based pixel-level object tracking that enables to reach the state-of-the-art results for both single and multiple object tracking.
- We conduct an extensive analysis to explore the factors contributing to our good results.
- We show that training a convnet for object tracking can be done with only few annotated frames. We hope these results will affect the trend towards even larger training sets, and popularize the design of trackers with lighter training needs.

10.2 PREVIOUS WORK ON SYNTHETIC DATA

Like our approach, previous works have also explored synthesizing training data. Synthetic renderings (Mayer *et al.*, 2016), video game environment (Richter *et al.*, 2016), mix-synthetic and real images (Varol *et al.*; Chen *et al.*, 2016c; Dosovitskiy *et al.*, 2015) have shown promise, but require task-appropriate 3d models. Compositing real world images provides more realistic results, and has shown promise for object detection (Georgakis *et al.*, 2017; Tang *et al.*, 2013), text localization (Gupta *et al.*, 2016) and pose estimation (Pishchulin *et al.*, 2012).

The closest work to ours is Park and Ramanan (2015), which also generates video-specific training data using the first frame annotations. They use human skeleton annotations to improve pose estimation, while we employ mask annotations to improve object tracking.

10.3 LUCIDTRACKER

Section 10.3.1 describes the network architecture used, and how RGB and optical flow information are fused to predict the next frame segmentation mask. Section 10.3.2 discusses different training modalities employed with the proposed object tracking system. In Section 10.4 we discuss the training data generation, and sections 10.5/10.6 report results for single/multiple object tracking.

10.3.1 Architecture

Approach. We model the pixel-level object tracking problem as a mask refinement task (mask: binary foreground/ background labelling of the image) based on appearance and motion cues. From frame $t-1$ to frame t the estimated mask M_{t-1} is propagated to frame t, and the new mask M_t is computed as a function of the

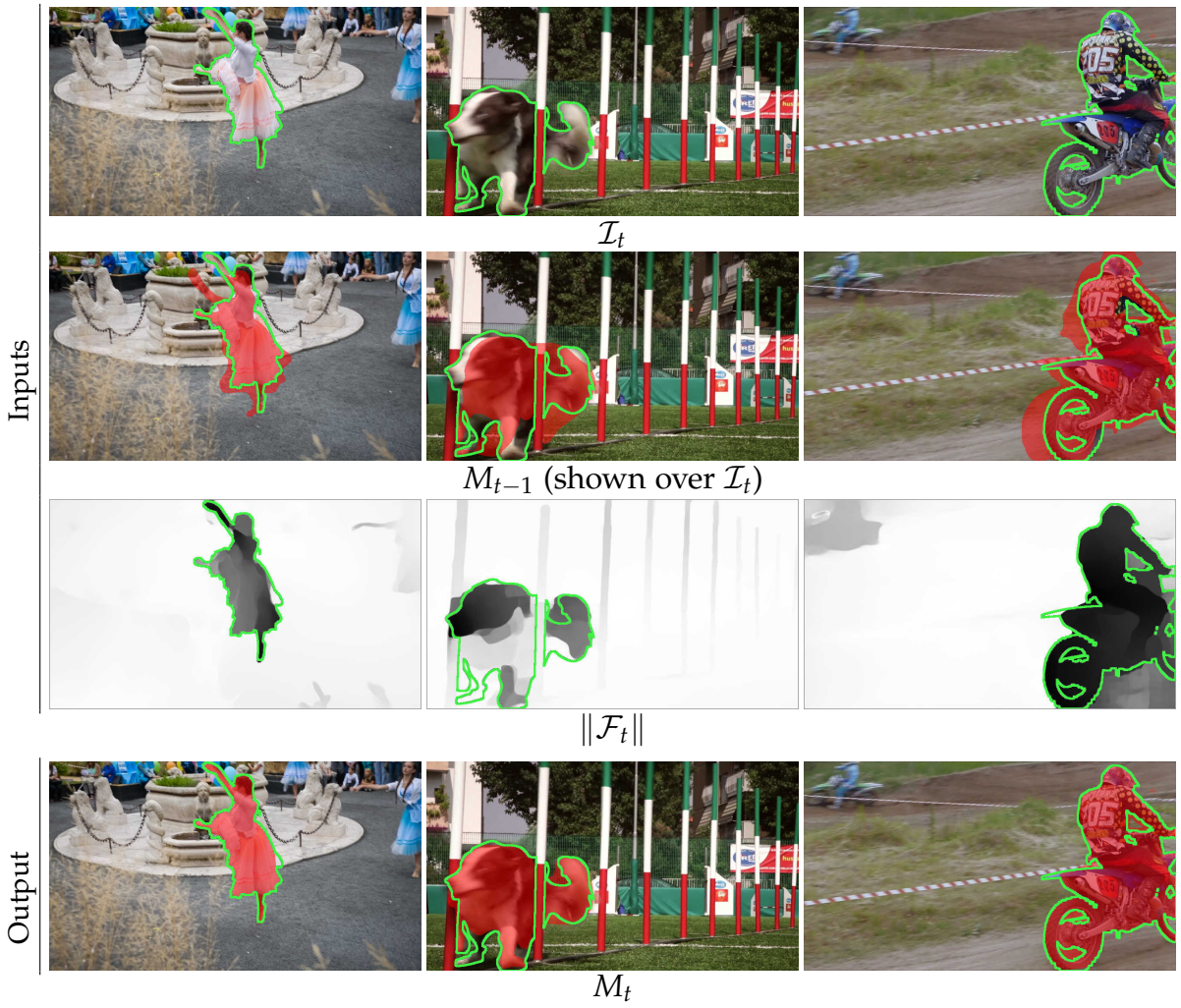


Figure 10.2: Data flow examples. $\mathcal{I}_t$, $\mathcal{F}_t$, M_{t-1} are the inputs, M_t is the resulting output. Green boundaries outline the ground truth segments. Red overlay indicates M_{t-1}, M_t.

previous mask, the new image $\mathcal{I}_t$, and the optical flow $\mathcal{F}_t$, i.e. $M_t = f(\mathcal{I}_t, \mathcal{F}_t, M_{t-1})$. Since objects have a tendency to move smoothly through space in time, there are little changes from frame to frame and mask M_{t-1} can be seen as a rough estimate of M_t. Thus we require our trained convnet to learn to refine rough masks into accurate masks. Fusing the complementary image $\mathcal{I}_t$ and motion $\mathcal{F}_t$ enables to exploits the information inherent to video and enables the model to segment well both static and moving objects.

Note that this approach is incremental, does a single forward pass over the video, and keeps no explicit model of the object appearance at frame t. In some experiments we adapt the model f per video, using the annotated first frame $\mathcal{I}_0$, M_0. However, in contrast to traditional techniques (Henriques *et al.*, 2012), this model is not updated while we process the video frames, thus the only state evolving along the video is the mask M_{t-1} itself.

First frame. In the video object tracking task the mask for the first frame M_0 is given. This is the standard protocol of the benchmarks considered in sections 10.5 & 10.6. If only a bounding box is available on the first frame, then the mask could be estimated using grabcut-like techniques (Rother *et al.*, 2004; Tang *et al.*, 2016).

RGB image $\mathcal{I}$. Typically a semantic labeller generates pixel-wise labels based on the input image (e.g. $M = g(\mathcal{I})$). We use an augmented semantic labeller with an input layer modified to accept 4 channels (RGB + previous mask) so as to generate outputs based on the previous mask estimate, e.g. $M_t = f_{\mathcal{I}}(\mathcal{I}_t, M_{t-1})$. Our approach is general and can leverage any existing semantic labelling architecture. We select the DeepLabv2 architecture with VGG base network (Chen *et al.*, 2016b), which is comparable to Jampani *et al.* (2016a); Caelles *et al.* (2017b); Perazzi *et al.* (2017); FusionSeg (Jain *et al.*, 2017) uses ResNet.

Optical flow $\mathcal{F}$. We use flow in two complementary ways. First, to obtain a better initial estimate of M_t we warp M_{t-1} using the flow $\mathcal{F}_t$: $M_t = f_{\mathcal{I}}(\mathcal{I}_t, w(M_{t-1}, \mathcal{F}_t))$. Second, we use flow as a direct source of information about the mask M_t. As can be seen in Figure 10.2, when the object is moving relative to background, the flow magnitude $\|\mathcal{F}_t\|$ provides a very reasonable estimate of the mask M_t. We thus consider using a convnet specifically for mask estimation from flow: $M_t = f_{\mathcal{F}}(\mathcal{F}_t, w(M_{t-1}, \mathcal{F}_t))$, and merge it with the image-only version by naive averaging

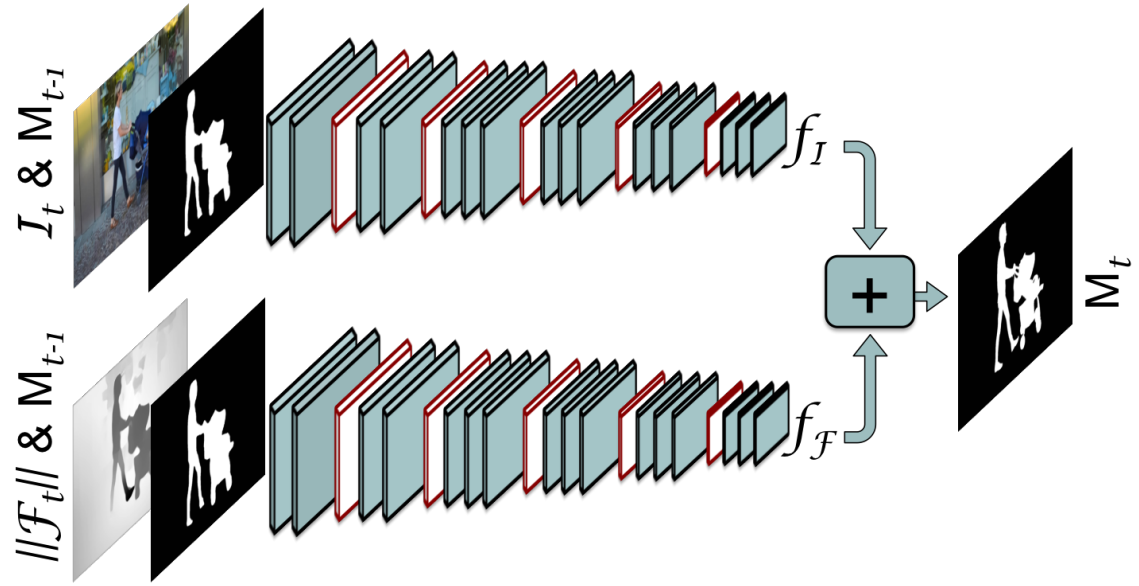
$$M_t = 0.5 \cdot f_{\mathcal{I}}(\mathcal{I}_t, \ldots) + 0.5 \cdot f_{\mathcal{F}}(\mathcal{F}_t, \ldots). \tag{10.1}$$

We use the state-of-the-art optical flow estimation method FlowNet2.0 (Ilg *et al.*, 2017), which itself is a convnet that computes $\mathcal{F}_t = h(\mathcal{I}_{t-1}, \mathcal{I}_t)$ and is trained on synthetic renderings of flying objects (Mayer *et al.*, 2016). For the optical flow magnitude computation we subtract the median motion for each frame, average the magnitude of the forward and backward flow and scale the values to $[0; 255]$, bringing it to the same range as RGB channels.

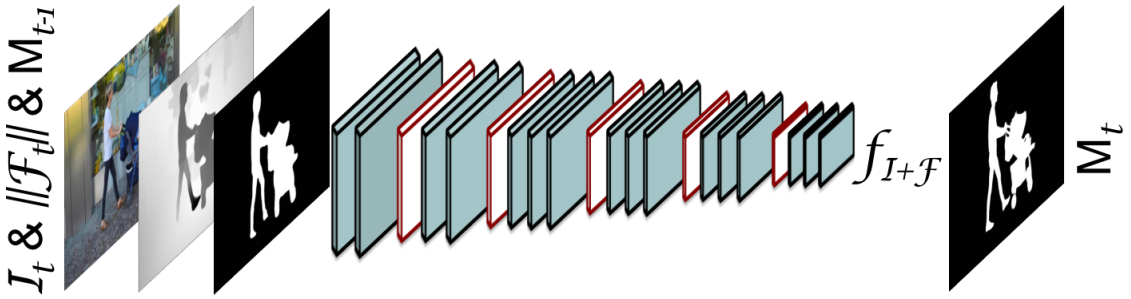
The loss function is the sum of cross-entropy terms over each pixel in the output map (all pixels are equally weighted). In our experiments $f_{\mathcal{I}}$ and $f_{\mathcal{F}}$ are trained independently, via some of the modalities listed in Section 10.3.2. Our two streams architecture is illustrated in Figure 10.3a.

We also explored expanding our network to accept 5 input channels (RGB + previous mask + flow magnitude) in one stream: $M_t = f_{\mathcal{I}+\mathcal{F}}(\mathcal{I}_t, \mathcal{F}_t, w(M_{t-1}, \mathcal{F}_t))$, but did not observe much difference in the performance compared to naive averaging, see experiments in Section 10.5.4.3. Our one stream architecture is illustrated in Figure 10.3b. One stream network is more affordable to train and allows to easily add extra input channels, e.g. providing additionally semantic information about objects.

Multiple objects. The proposed framework can easily be extended to multiple object tracking. Instead of having one additional channel for the previous frame mask we provide the mask for each object in a separate channel, expanding



(a) Two streams architecture, where image $\mathcal{I}_t$ and optical flow information $\|\mathcal{F}_t\|$ are used to update mask M_{t-1} into M_t. See equation 10.1.



(b) One stream architecture, where 5 input channels: image $\mathcal{I}_t$, optical flow information $\|\mathcal{F}_t\|$ and mask M_{t-1} are used to estimate mask M_t.

Figure 10.3: Overview of the proposed one and two streams architectures. See §10.3.1.

the network to accept $3 + N$ input channels (RGB + N object masks): $M_t = f_{\mathcal{I}}\left(\mathcal{I}_t, w(M^1_{t-1}, \mathcal{F}_t), ..., w(M^N_{t-1}, \mathcal{F}_t)\right)$, where N is the number of objects annotated on the first frame.

For multiple object tracking task we employ a one-stream architecture for the experiments, using optical flow $\mathcal{F}$ and semantic segmentation $\mathcal{S}$ as additional input channels: $M_t{=}f_{\mathcal{I}+\mathcal{F}+\mathcal{S}}\left(\mathcal{I}_t, \mathcal{F}_t, \mathcal{S}_t, w(M^1_{t-1}, \mathcal{F}_t), ..., w(M^N_{t-1}, \mathcal{F}_t)\right)$. This allows to leverage the appearance model with semantic priors and motion information. See Figure 10.4 for an illustration.

We use the state-of-the-art semantic segmentation method PSPNet (Zhao *et al.*,

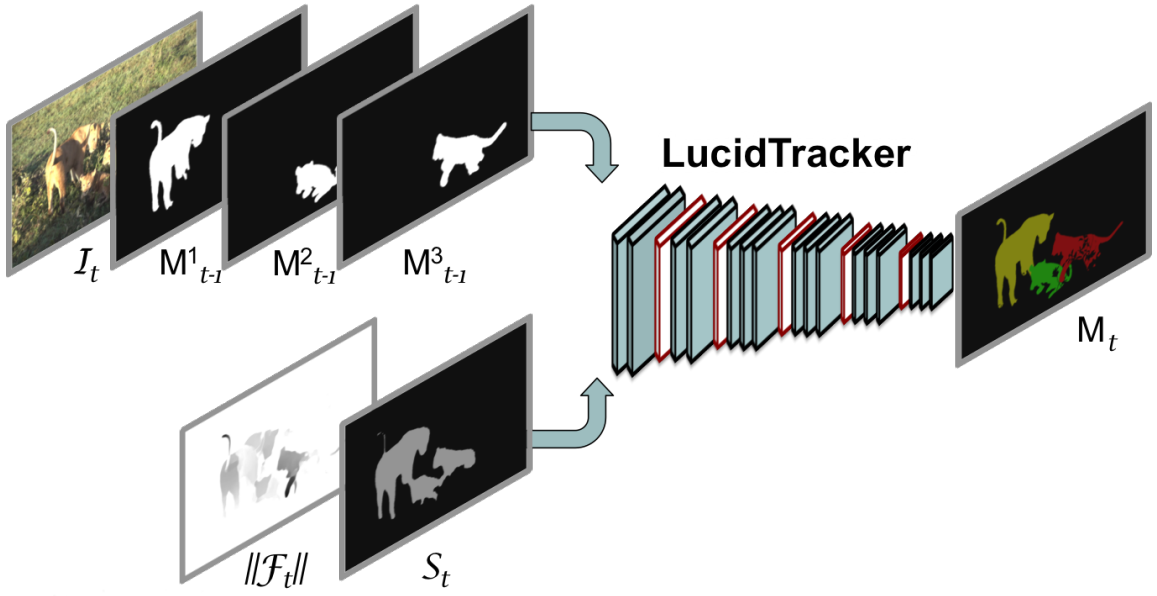


Figure 10.4: Extension of LucidTracker to multiple objects. The previous frame mask for each object is provided in a separate channel. We additionally explore using optical flow $\mathcal{F}$ and semantic segmentation $\mathcal{S}$ as additional inputs. See §10.3.1.

2016), which itself is a convnet that computes a pixel-level labelling $\mathcal{S}_t = h(\mathcal{I}_t)$ and is trained on Pascal VOC12 (Everingham *et al.*, 2015). Pascal VOC12 annotates 20 categories. Since we consider all semantic classes, $\mathcal{S}_t$ can also provide information about unknown category instances by describing them as a spatial mixture of known ones (e.g. a sea lion might looks like a dog torso, and the head of cat). As long as the predictions are consistent through time, $\mathcal{S}_t$ will provide a useful cue for tracking. Note that we only use $\mathcal{S}_t$ for the multi-object tracking challenge, discussed in Section 10.6. In the same way as for the optical flow we scale $\mathcal{S}_t$ to bring all the channels to the same range.

We additionally experiment with ensembles of different variants, that allows to make the system more robust to the challenges inherent in videos. For our main results for multiple object tracking task we consider the ensemble of four models: M_t=$0.25 \cdot (f_{\mathcal{I}+\mathcal{F}+\mathcal{S}} + f_{\mathcal{I}+\mathcal{F}} + f_{\mathcal{I}+\mathcal{S}} + f_{\mathcal{I}})$, where we merge the outputs of the models by naive averaging. See Section 10.6 for more details.

Post-processing. As a final stage of our pipeline, we refine per-frame t the generated mask M_t using DenseCRF (Krähenbühl and Koltun, 2011). This adjusts small image details that the network might not be able to handle. It is known by practitioners that DenseCRF is quite sensitive to its parameters and can easily worsen results. We will use our lucid dreams to handle per-dataset CRF-tuning too, see Section 10.3.2. We refer to our full $f_{\mathcal{I}+\mathcal{F}}$ system as `LucidTracker`, and as `LucidTracker`$^-$ when no post-processing is used. The usage of S_t or model ensemble will be explicitly stated.

10.3.2 Training modalities

Multiple modalities are available to train a tracker. **Training-free** approaches (e.g. BVS (Maerki *et al.*, 2016), SVT (Wang and Shen, 2017)) are fully hand-crafted systems with hand-tuned parameters, and thus do not require training data. They can be used as-is over different datasets. Supervised methods can also be trained to generate a **dataset-agnostic** model that can be applied over different datasets. Instead of using a fixed model for all cases, it is also possible to obtain specialized **per-dataset** models, either via self-supervision (Wang and Gupta, 2015; Pathak *et al.*, 2016; Yu *et al.*, 2016; Zhu *et al.*, 2017b) or by using the first frame annotation of each video in the dataset as training/tuning set. Finally, inspired by traditional tracking techniques, we also consider adapting the model weights to the specific video at hand, thus obtaining **per-video** models. Section 10.5 reports new results over these four training modalities (training-free, dataset-agnostic, per-dataset, and per-video).

Our `LucidTracker` obtains best results when first pre-trained on ImageNet, then trained per-dataset using all data from first frame annotations together, and finally fine-tuned per-video for each evaluated sequence. The post-processing DenseCRF stage is automatically tuned per-dataset. The experimental section 10.5 details the effect of these training stages. Interestingly, we can obtain reasonable performance even when training from only a single annotated frame (without ImageNet pre-training).

Unless otherwise stated, we fine-tune per-video models relying solely on the first frame $\mathcal{I}_0$ and its annotation M_0. This is in contrast to traditional techniques (Henriques *et al.*, 2012; Breitenstein *et al.*, 2009; Kristan *et al.*, 2014) which would update the appearance model at each frame $\mathcal{I}_t$.

10.4 LUCID DATA DREAMING

To train the function f one would think of using ground truth data for M_{t-1} and M_t (like Bertinetto *et al.* (2016); Caelles *et al.* (2017b); Held *et al.* (2016)); however, such data is expensive to annotate and rare. Caelles *et al.* (2017b) thus train on a set of 30 videos ($\sim 2k$ frames) and requires the model to transfer across multiple tests sets. Perazzi *et al.* (2017) side-step the need for consecutive frames by generating synthetic masks M_{t-1} from a saliency dataset of $\sim 10k$ images with their corresponding mask M_t . We propose a new data generation strategy to reach better results using only ~ 100 individual training frames.

Ideally training data should be as similar as possible to the test data, even subtle differences may affect quality (e.g. training on static images for testing on videos under-performs (Tang *et al.*, 2012)). To ensure our training data is in-domain, we propose to generate it by synthesizing samples from the provided annotated frame (first frame) in each target video. This is akin to "lucid dreaming" as we intentionally "dream" the desired data by creating sample images that are plausible hypothetical future frames of the video. The outcome of this process is a large set of frame pairs

in the target domain (2.5k pairs per annotation) with known optical flow and mask annotations, see Figure 10.5.

Synthesis process. The target domain for a tracker is the set of future frames of the given video. Traditional data augmentation via small image perturbation is insufficient to cover the expect variations across time, thus a task specific strategy is needed. Across the video the tracked object might change in illumination, deform, translate, be occluded, show different point of views, and evolve on top of a dynamic background. All of these aspects should be captured when synthesizing future frames. We achieve this by cutting-out the foreground object, in-painting the background, perturbing both foreground and background, and finally recomposing the scene. This process is applied twice with randomly sampled transformation parameters, resulting in a pair of frames ($\mathcal{I}_{\tau-1}$, $\mathcal{I}_\tau$) with known pixel-level ground-truth mask annotations ($M_{\tau-1}$, M_τ), optical flow $\mathcal{F}_\tau$, and occlusion regions. The object position in $\mathcal{I}_\tau$ is uniformly sampled, but the changes between $\mathcal{I}_{\tau-1}$, $\mathcal{I}_\tau$ are kept small to mimic the usual evolution between consecutive frames.

In more details, starting from an annotated image:

1. *Illumination changes:* we globally modify the image by randomly altering saturation S and value V (from HSV colour space) via $x' = a \cdot x^b + c$, where $a \in 1 \pm 0.05$, $b \in 1 \pm 0.3$, and $c \in \pm 0.07$.

2. *Fg/Bg split:* the foreground object is removed from the image $\mathcal{I}_0$ and a background image is created by inpainting the cut-out area (Criminisi *et al.*, 2004).

3. *Object motion:* we simulate motion and shape deformations by applying global translation as well as affine and non-rigid deformations to the foreground object. For $\mathcal{I}_{\tau-1}$ the object is placed at any location within the image with a uniform distribution, and in $\mathcal{I}_\tau$ with a translation of $\pm 10\%$ of the object size relative to $\tau - 1$. In both frames we apply random rotation $\pm 30^\circ$, scaling $\pm 15\%$ and thin-plate splines deformations (Bookstein, 1989) of $\pm 10\%$ of the object size.

4. *Camera motion:* We additionally transform the background using affine deformations to simulate camera view changes. We apply here random translation, rotation, and scaling within the same ranges as for the foreground object.

5. *Fg/Bg merge:* finally ($\mathcal{I}_{\tau-1}$, $\mathcal{I}_\tau$) are composed by blending the perturbed foreground with the perturbed background using Poisson matting (Sun *et al.*, 2004). Using the known transformation parameters we also synthesize ground-truth pixel-level mask annotations ($M_{\tau-1}$, M_τ) and optical flow $\mathcal{F}_\tau$.

Figure 10.5 shows example results. Albeit our approach does not capture appearance changes due to point of view, occlusions, nor shadows, we see that already this rough modelling is effective to train our tracking models.

The number of synthesized images can be arbitrarily large. We generate 2.5k pairs per annotated video frame. This training data is, by design, in-domain with

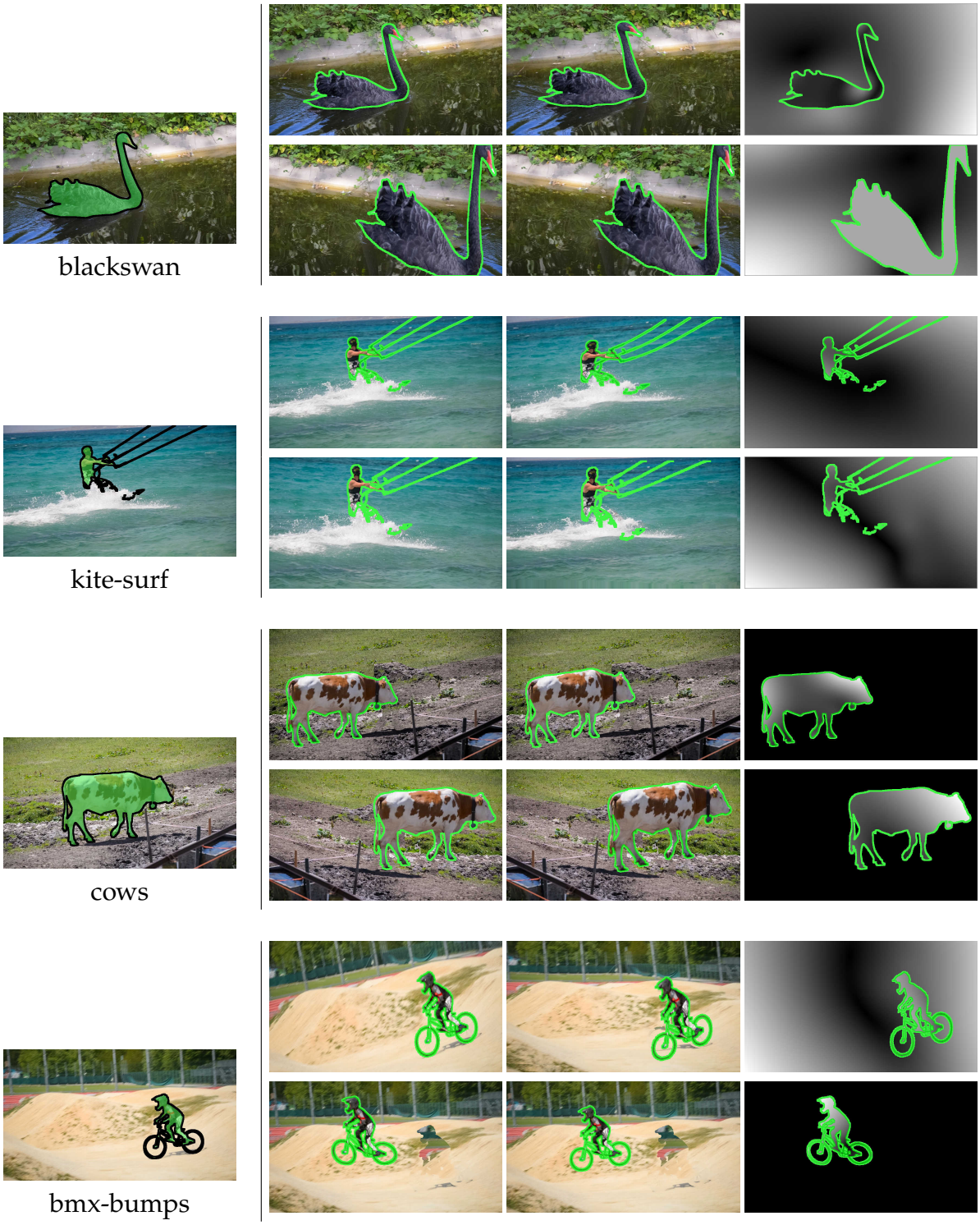


Figure 10.5: Lucid data dreaming examples. From one annotated frame we generate pairs of images ($\mathcal{I}_{\tau-1}$, $\mathcal{I}_\tau$) that are plausible future video frames, with known optical flow ($\mathcal{F}_\tau$) and masks (green boundaries). Note the inpainted background and foreground/background deformations.

(a) Original image $\mathcal{I}_0$ and mask annotation M_0

(b) Generated image $\mathcal{I}_\tau$ and mask M_τ

(c) Generated flow magnitude $\|\mathcal{F}_\tau\|$

Figure 10.6: Lucid data dreaming examples with multiple objects. From one annotated frame we generate a plausible future video frame ($\mathcal{I}_\tau$), with known optical flow ($\mathcal{F}_\tau$) and mask (M_τ).

regard of the target video. The experimental section 10.5 shows that this strategy is more effective than using thousands of manually annotated images from close-by domains.

The same strategy for data synthesis can be employed for multiple object tracking task. Instead of manipulating a single object we handle multiple ones at the same time, applying independent transformations to each of them. We model occlusion between objects by adding a random depth ordering obtaining both partial and full occlusions in the training set. Including occlusions in the lucid dreams allows to better handle plausible interactions of objects in the future frames. See Figure 10.6 an example illustration.

10.5 SINGLE OBJECT TRACKING RESULTS

We present here a detailed empirical evaluation on three different datasets for single object tracking task: given a first frame labelled with the foreground object mask, the goal is to find the corresponding object pixels in future frames. (Section 10.6 will discuss the multiple objects case.)

10.5.1 Experimental setup

Datasets. We evaluate our method on three video object segmentation datasets: DAVIS_{16} (Perazzi *et al.*, 2016), YouTubeObjects (Prest *et al.*, 2012; Jain and Grauman, 2014), and $\text{SegTrack}_{\text{v2}}$ (Li *et al.*, 2013). The goal is to track an object through all video frames given a foreground object mask in the first frame. These three datasets provide diverse challenges with a mix of high and low resolution web videos, single or multiple salient objects per video, videos with flocks of similar looking instances, longer ($\sim$400) and shorter ($\sim$10) sequences, as well as the usual tracking challenges such as occlusion, fast motion, illumination, view point changes, elastic deformation, etc.

The DAVIS_{16} (Perazzi *et al.*, 2016) video segmentation benchmark consists of 50 full-HD videos of diverse object categories with all frames annotated with pixel-level accuracy, where one single or two connected moving objects are separated from the background. The number of frames in each video varies from 25 to 104.

YouTubeObjects (Prest *et al.*, 2012; Jain and Grauman, 2014) includes web videos from 10 object categories. We use the subset of 126 video sequences with mask annotations provided by Jain and Grauman (2014) for evaluation, where one single object or a group of objects of the same category are separated from the background. In contrast to DAVIS_{16} these videos have a mix of static and moving objects. The number of frames in each video ranges from 2 to 401.

$\text{SegTrack}_{\text{v2}}$(Li *et al.*, 2013) consists of 14 videos with multiple object annotations for each frame. For videos with multiple objects each object is treated as a separate problem, resulting in 24 sequences. The length of each video varies from 21 to 279 frames. The images in this dataset have low resolution and some compression artefacts, making it hard to track the object based on its appearance.

The main experimental work is done on DAVIS_{16}, since it is the largest densely annotated dataset out of the three, and provides high quality/high resolution data. The videos for this dataset were chosen to represent diverse challenges, making it a good experimental playground. We additionally report on two other datasets as complementary test set results.

Evaluation metric. To measure the accuracy of video object tracking we use the mean intersection-over-union overlap (mIoU) between the per-frame ground truth object mask and the predicted segmentation, averaged across all video sequences. We have noticed disparate evaluation procedures used in previous work, and we report here a unified evaluation across datasets. When possible, we re-evaluated certain methods using results provided by their authors. For all three datasets we follow the DAVIS_{16} evaluation protocol, excluding the first frame from evaluation and using all other frames from the video sequences, independent of object presence in the frame.

Training details. For training all the models we use SGD with mini-batches of 10 images and a fixed learning policy with initial learning rate of 10^{-3}. The momentum

Figure 10.7: LucidTracker single object tracking qualitative results. Frames sampled along the video duration (e.g. 50%: video middle point). Our model is robust to various challenges, such as view changes, fast motion, shape deformations, and out-of-view scenarios.

and weight decay are set to 0.9 and $5 \cdot 10^{-4}$, respectively.

Models using pre-training are initialized with weights trained for image classification on ImageNet (Simonyan and Zisserman, 2015). We then train per-dataset for 40k iterations with the RGB+Mask branch $f_{\mathcal{I}}$ and for 20k iterations for the Flow+Mask $f_{\mathcal{F}}$ branch. When using a single stream architecture (Section 10.5.4.3), we use 40k iterations. Models without ImageNet pre-training are initialized using the "Xavier" strategy (Glorot and Bengio, 2010). The per-dataset training needs to be longer, using 100k iterations for the $f_{\mathcal{I}}$ branch and 40k iterations for the $f_{\mathcal{F}}$ branch. For per-video fine-tuning 2k iterations are used for $f_{\mathcal{I}}$. To keep computing cost lower, the $f_{\mathcal{F}}$ branch is kept fix across videos. All training parameters are chosen based on DAVIS_{16} results. We use identical parameters on YouTubeObjects and $\text{SegTrack}_{\text{v2}}$, showing the generalization of our approach.

It takes ~3.5h to obtain each per-video model, including data generation, per-dataset training, per-video fine-tuning and per-dataset grid search of CRF parameters (averaged over DAVIS_{16}, amortising the per-dataset training time over all videos). At test time our LucidTracker runs at ~5s per frame, including the optical flow estimation with FlowNet2.0 (Ilg *et al.*, 2017) (~0.5s) and CRF post-processing (Krähenbühl and Koltun, 2011) (~2s).

10.5.2 Key results

Table 10.1 presents our main result and compares it to previous work. Our full system, `LucidTracker`, provides the best tracking quality across three datasets while being trained on each dataset using only one frame per video (50 frames for DAVIS_{16}, 126 for YouTubeObjects, 24 for $\text{SegTrack}_{\text{v2}}$), which is $20\times \sim 100\times$ less than the top competing methods. Ours is the first method to reach > 75 mIoU on all three datasets.

Oracles and baselines. Grabcut oracle computes grabcut (Rother *et al.*, 2004) using the ground truth bounding boxes (box oracle). This oracle indicates that on the considered datasets separating foreground from background is not easy, even if a perfect box-level tracker was available.

We provide three additional baselines. "Saliency" corresponds to using the generic (training-free) saliency method EQCut (Aytekin *et al.*, 2015) over the RGB image $\mathcal{I}_t$. "Flow saliency" does the same, but over the optical flow magnitude $\|\mathcal{F}_t\|$. Results indicate that the objects being tracked are not particularly salient in the image. On DAVIS_{16} motion saliency is a strong signal but not on the other two datasets. Saliency methods ignore the first frame annotation provided for the tracking task. We also consider the "Mask warping" baseline which uses optical flow to propagate the mask estimate from t to $t+1$ via simple warping $M_t = w(M_{t-1}, \mathcal{F}_t)$. The bad results of this baseline indicate that the high quality flow (Ilg *et al.*, 2017) that we use is by itself insufficient to solve the tracking task, and that indeed our proposed convnet does the heavy lifting. The large fluctuation of the relative baseline results across the three datasets empirically confirms that each of them presents unique

	Method	# training images	Flow $\mathcal{F}$	Dataset, mIoU DAVIS$_{16}$	YoutbObjs	SegTrck$_{v2}$
	Box oracle (Perazzi *et al.*, 2017)	0	✗	45.1	55.3	56.1
	Grabcut oracle (Perazzi *et al.*, 2017)	0	✗	67.3	67.6	74.2
Ignores 1st frame annotation	Saliency	0	✗	32.7	40.7	22.2
	NLC (Faktor and Irani, 2014)	0	✓	64.1	-	-
	TRS (Xiao and Lee, 2016)	0	✓	-	-	69.1
	MP-Net (Tokmakov *et al.*, 2017a)	~22.5k	✓	69.7	-	-
	Flow saliency	0	✓	70.7	36.3	35.9
	FusionSeg (Jain *et al.*, 2017)	~95k	✓	71.5	67.9	-
Uses 1st frame annotation	Mask warping	0	✓	32.1	43.2	42.0
	FCP (Perazzi *et al.*, 2015)	0	✓	63.1	-	-
	BVS (Maerki *et al.*, 2016)	0	✗	66.5	59.7	58.4
	N15 (Nagaraja *et al.*, 2015)	0	✓	-	-	69.6
	ObjFlow (Tsai *et al.*, 2016)	0	✓	71.1	70.1	67.5
	STV (Wang and Shen, 2017)	0	✓	73.6	-	-
	VPN (Jampani *et al.*, 2016a)	~2.3k	✗	*75.0*	-	-
	OSVOS (Caelles *et al.*, 2017b)	~2.3k	✗	*79.8*	72.5	65.4
	MaskTrack (Perazzi *et al.*, 2017)	~11k	✓	80.3	72.6	70.3
	LucidTracker	**24~126**	✓	**84.8**	**76.2**	**77.6**

Table 10.1: Comparison of segment tracking results across three datasets. Numbers in italic are reported on subsets of DAVIS$_{16}$. Our LucidTracker consistently improves over previous results, see §10.5.2.

challenges.

Comparison. Compared to flow propagation methods such as BVS, N15, ObjFlow, and STV, we obtain better results because we build per-video a stronger appearance model of the tracked object (embodied in the fine-tuned model). Compared to convnet learning methods such as VPN, OSVOS, MaskTrack, we require significantly less training data, yet obtain better results.

Figure 10.7 provides qualitative results of `LucidTracker` across three different datasets. Our system is robust to various challenges present in videos. It handles well camera view changes, fast motion, object shape deformation, out-of-view scenarios, multiple similar looking objects and even low quality video. We provide a detailed error analysis in Section 10.5.5.

Conclusion. We show that using less training data, does not necessarily lead to poorer results. We report top results for this task while using only 24$\sim$126 training

Variant	$\mathcal{I}$	$\mathcal{F}$	warp. w	per-video fine-tun. $\mathcal{F}$	Dataset, mIoU DAVIS$_{16}$	YoutbObjs	SegTrck$_{v2}$
LucidTracker	✓	✓	✓	✗	**84.8**	**76.2**	**77.6**
LucidTracker$^-$	✓	✓	✓	✗	83.7	76.2	76.8
No warping	✓	✓	✗	✗	82.0	74.6	70.5
No OF	✓	✗	✗	✗	78.0	74.7	61.8
OF only	✗	✓	✓	✗	74.5	43.1	55.8

Table 10.2: Ablation study of flow ingredients. Flow complements image only results, with large fluctuations across datasets. See §10.5.3.1.

Variant	Optical flow	Dataset, mIoU DAVIS$_{16}$	YoutbObjs	SegTrck$_{v2}$
LucidTracker$^-$	FlowNet2.0	**83.7**	**76.2**	**76.8**
	EpicFlow	80.2	71.3	67.0
	No flow	78.0	74.7	61.8
No ImageNet pre-training	FlowNet2.0	82.0	74.3	71.2
	EpicFlow	80.0	72.3	68.8
	No flow	76.7	71.4	63.0

Table 10.3: Effect of optical flow estimation.

frames.

10.5.3 Ablation studies

In this section we explore in more details how the different ingredients contribute to our results.

10.5.3.1 *Effect of optical flow*

Table 10.2 shows the effect of optical flow on `LucidTracker` results. Comparing our full system to the "No OF" row, we see that the effect of optical flow varies across datasets, from minor improvement in YouTubeObjects, to major difference in SegTrack$_{v2}$. In this last dataset, using mask warping is particularly useful too. We additionally explored tuning the optical flow stream per-video, which resulted in a minor improvement (83.7$\rightarrow$83.9 mIoU on DAVIS$_{16}$).

OSVOS (Caelles *et al.*, 2017b) also does not use optical flow, but instead uses a

Variant	ImgNet pre-train.	per-dataset training	per-video fine-tun.	Dataset, mIoU DAVIS$_{16}$	YoutbObjs	SegTrck$_{v2}$
LucidTracker$^-$	✓	✓	✓	**83.7**	**76.2**	**76.8**
(no ImgNet)	✗	✓	✓	82.0	74.3	71.2
No per-video	✓	✓	✗	82.7	72.3	71.9
tuning	✗	✓	✗	78.4	69.7	68.2
Only per-	✓	✗	✓	79.4	-	70.4
-video tuning	✗	✗	✓	*80.5*	-	*66.8*

Table 10.4: Ablation study of training modalities. ImageNet pre-training and per-video tuning provide additional improvement over per-dataset training. Even with one frame annotation for only per-video tuning we obtain good performance. See §10.5.3.2.

per-frame mask post-processing based on a boundary detector (trained on further external data), which provides $\sim$2 percent point gain. Accounting for this, our "No OF" (and no CRF) result matches theirs on DAVIS$_{16}$ and YouTubeObjects despite using significantly less training data (see Table 10.1, e.g. $79.8 - 2 \approx 78.0$ on DAVIS$_{16}$).

Table 10.3 shows the effect of using different optical flow estimation methods. For `LucidTracker` results, FlowNet2.0 (Ilg *et al.*, 2017) was employed. We also explored using EpicFlow (Revaud *et al.*, 2015), as in Perazzi *et al.* (2017). Table 10.3 indicates that employing a robust optical flow estimation across datasets is crucial to the performance (FlowNet2.0 provides $\sim 1.5 - 15$ points gain on each dataset). We found EpicFlow to be brittle when going across different datasets, providing improvement for DAVIS$_{16}$ and SegTrack$_{v2}$ ($\sim 2 - 5$ points gain), but underperforming for YouTubeObjects (74.7$\rightarrow$71.3 mIoU).

Conclusion. The results show that flow provides a complementary signal to RGB image only and having a robust optical flow estimation across datasets is crucial. Despite its simplicity our fusion strategy ($f_{\mathcal{I}} + f_{\mathcal{F}}$) provides gains on all datasets, and leads to competitive results.

10.5.3.2 *Effect of training modalities*

Table 10.4 compares the effect of different ingredients in the `LucidTracker`$^-$ training. Results are obtained using RGB and flow, with warping, no CRF; $M_t = f(\mathcal{I}_t, w(M_{t-1}, \mathcal{F}_t))$. We see that ImageNet pre-training does provide $2 \sim 5$ percent point improvement (depending on the dataset of interest; e.g. 82.0$\rightarrow$83.7 mIoU on DAVIS$_{16}$). Per-video fine-tuning (after doing per-dataset training) provides an additional $1 \sim 2$ percent point gain (e.g. 82.7$\rightarrow$83.7 mIoU on DAVIS$_{16}$). Both ingredients clearly contribute to the tracking results.

Method	CRF parameters	Dataset, mIoU		
		$DAVIS_{16}$	YoutbObjs	$SegTrck_{v2}$
LucidTracker$^-$	-	83.7	76.2	76.8
LucidTracker	default	84.2	75.5	72.2
LucidTracker	tuned per-dataset	**84.8**	**76.2**	**77.6**

Table 10.5: Effect of CRF tuning. Without per-dataset tuning DenseCRF will underperform.

In the bottom row ("only per-video tuning"), the model is trained per-video without ImageNet pre-training nor per-dataset training, i.e. using a *single annotated training frame*. Our network is based on VGG16 (Chen *et al.*, 2016b) and contains $\sim 20M$ parameters, all effectively learnt from a single annotated image that is augmented to become $2.5k$ training samples (see Section 10.4). Even with such minimal amount of training data, we still obtain a surprisingly good performance (compare 80.5 on $DAVIS_{16}$ to others in Table 10.1). This shows how effective is, by itself, the proposed training strategy based on lucid dreaming of the data. Note that training a model using only per-video tuning takes about one full GPU day per video sequence; making these results insightful but not decidedly practical.

Preliminary experiments evaluating on $DAVIS_{16}$ the impact of the different ingredients of our lucid dreaming data generation showed, depending on the exact setup, $3\sim10$ percent mIoU points fluctuations between a basic version (e.g. without non-rigid deformations nor scene re-composition) and the full synthesis process described in Section 10.4. Having a sophisticated data generation process directly impacts the tracking quality.

Conclusion. Both ImageNet pre-training and per-video tuning of the models provide complementary gains over the default per-dataset training. Per-video training by itself, despite using a single annotated frame, provides already much of the needed information for the tracking task.

10.5.3.3 *Effect of CRF tuning*

As a final stage of our pipeline, we refine the generated mask using DenseCRF (Krähenbühl and Koltun, 2011) per frame. This captures small image details that the network might have missed. It is known by practitioners that DenseCRF is quite sensitive to its parameters and can easily worsen results. We use our lucid dreams to enable automatic per-dataset CRF-tuning.

Following Chen *et al.* (2016b) we employ grid search scheme for tuning CRF parameters. Once the per-dataset tracking model is trained, we apply it over a subset of its training set (5 random images from the lucid dreams per video sequence), apply DenseCRF with the given parameters over this output, and then compare to

Figure 10.8: Effect of CRF tuning. The shown DAVIS_{16} videos have the highest margin between with and without CRF post-processing (based on mIoU over the video).

the lucid dream ground truth.

The impact of the tuned parameter of DenseCRF post-processing is shown in Table 10.5 and Figure 10.8. Table 10.5 indicates that without per-dataset tuning DenseCRF is under-performing. Our automated tuning procedure allows to obtain consistent gains without the need for case-by-case manual tuning.

Conclusion. Using default DenseCRF parameters will degrade performance. Our lucid dreams enable per-dataset CRF-tuning which allows to further improve the results.

10.5.4 Additional experiments

Other than adding or removing ingredients, as in Section 10.5.3, we also want to understand how the training data itself affects the obtained results.

10.5.4.1 *Generalization across videos*

Table 10.6 explores the effect of tracking quality as a function of the number of training samples. To see more directly the training data effects we use a base model with RGB image $\mathcal{I}_t$ only (no flow $\mathcal{F}$, no CRF), and per-dataset training (no ImageNet pre-training, no per-video fine-tuning). We evaluate on two disjoint subsets of 15 DAVIS_{16} videos each, where the first frames for per-dataset training are taken from only one subset. The reported numbers are thus comparable within Table 10.6, but not across to the other tables in the chapter. Table 10.6 reports results with varying

Training set	# training videos	# frames per video	mIoU
Includes 1st frames from test set	1	1	78.3
	2	1	75.4
	15	1	68.7
	30	1	65.4
	30	2	74.3
Excludes 1st frames from test set	2	1	11.6
	15	1	36.4
	30	1	41.7
	30	2	48.4

Table 10.6: Varying the number of training videos. A smaller training set closer to the target domain is better than a larger one. See Section 10.5.4.1.

number of training videos and with/without including the first frames of each test video for per-dataset training. When excluding the test set first frames, the image frames used for training are separate from the test videos; and we are thus operating across (related) domains. When including the test set first frames, we operate in the usual LucidTracker mode, where the first frame from each test video is used to build the per-dataset training set.

Comparing the top and bottom parts of the table, we see that when the annotated images from the test set videos are not included, tracking quality drops drastically (e.g. 68.7 $\rightarrow$ 36.4 mIoU). Conversely, on subset of videos for which the first frame annotation is used for training, the quality is much higher and improves as the training samples become more and more specific (in-domain) to the target video (65.4 $\rightarrow$ 78.3 mIoU). Adding extra videos for training does not improve the performance. It is better (68.7 $\rightarrow$ 78.3 mIoU) to have 15 models each trained and evaluated on a single video (row top-1-1) than having one model trained over 15 test videos (row top-15-1). Training with an additional frame from each video (we added the last frame of each train video) significantly boosts the resulting within-video quality (e.g. row top-30-2 65.4 $\rightarrow$ 74.3 mIoU), because the training samples cover better the test domain.

Conclusion. These results show that, when using RGB information ($\mathcal{I}_t$), increasing the number of training videos *does not* improve the resulting quality of our system. Even within a dataset, properly using the training sample(s) from within each video matters more than collecting more videos to build a larger training set.

Training set	Dataset, mIoU			Mean
	DAVIS_{16}	YoutbObjs	$\text{SegTrck}_{\mathbf{v2}}$	
DAVIS_{16}	<u>80.9</u>	50.9	46.9	59.6
YoutbObjs	*67.0*	<u>71.5</u>	*52.0*	63.5
$\text{SegTrack}_{\mathbf{v2}}$	56.0	*52.2*	<u>66.4</u>	58.2
Best	**80.9**	**71.5**	**66.4**	**72.9**
Second best	67.0	52.2	52.0	57.1
All-in-one	71.9	70.7	60.8	67.8

Table 10.7: Generalization across datasets. Results with underline are the best per dataset, and in italic are the second best per dataset (ignoring all-in-one setup). We observe a significant quality gap between training from the target videos, versus training from other datasets; see §10.5.4.2.

10.5.4.2 *Generalization across datasets*

Section 10.5.4.1 has explored the effect of changing the volume of training data within one dataset, Table 10.7 compares results when using different datasets for training. Results are obtained using a base model with RGB and flow ($M_t = f\,(\mathcal{I}_t,\, M_{t-1})$, no warping, no CRF), ImageNet pre-training, per-dataset training, and no per-video tuning to accentuate the effect of the training dataset.

The best performance is obtained when training on the first frames of the target set. There is a noticeable $\sim$10 percent points drop when moving to the second best choice (e.g. $80.9 \rightarrow 67.0$ for DAVIS_{16}). Interestingly, when putting all the datasets together for training ("all-in-one" row, a dataset-agnostic model) the results degrade, reinforcing the idea that "just adding more data" does not automatically make the performance better.

Conclusion. Best results are obtained when using training data that focuses on the test video sequences, using similar datasets or combining multiple datasets degrades the performance for our system.

10.5.4.3 *Experimenting with the convnet architecture*

Section 10.3.1 and Figure 10.3 described two possible architectures to handle $\mathcal{I}_t$ and $\mathcal{F}_t$. Previous experiments are all based on the two streams architecture.

Table 10.8 compares two streams versus one stream, where the network to accepts 5 input channels (RGB + previous mask + flow magnitude) in one stream: $M_t = f_{\mathcal{I}+\mathcal{F}}\,(\mathcal{I}_t,\, \mathcal{F}_t,\, w(M_{t-1},\, \mathcal{F}_t))$. Results are obtained using a base model with RGB and optical flow (no warping, no CRF), ImageNet pre-training, per-dataset training, and no per-video tuning. We observe that both one stream and two stream architecture with naive averaging perform on par. Using a one stream network makes

Architecture	ImgNet pre-train.	per-dataset training	per-video fine-tun.	$DAVIS_{16}$ mIoU
two streams	✓	✓	✗	**80.9**
one stream	✓	✓	✗	80.3

Table 10.8: Experimenting with the convnet architecture. See §10.5.4.3.

the training more affordable and allows more easily to expand the architecture with additional input channels.

Conclusion. One stream network performs as well as a network with two streams. We will use the one stream architecture in Section 10.6.

10.5.5 Error analysis

Table 10.9 presents an expanded evaluation on $DAVIS_{16}$ using evaluation metrics proposed in Perazzi *et al.* (2016). Three measures are used: region similarity in terms of intersection over union (J), contour accuracy (F, higher is better), and temporal instability of the masks (T, lower is better). We outperform the competitive methods (Perazzi *et al.*, 2017; Caelles *et al.*, 2017b) on all three measures.

Table 10.10 reports attribute based evaluation on $DAVIS_{16}$. `LucidTracker` is best on 13 out of 15 video attribute categories. This shows that `LucidTracker` can handle various video challenges present in $DAVIS_{16}$.

We present the per-sequence and per-frame results of `LucidTracker` over $DAVIS_{16}$ in Figure 10.9. On the whole we observe that the proposed approach is quite robust, most video sequences reach an average performance above 80 mIoU. However, by looking at per-frame results for each video (blue dots in Figure 10.9) one can see several frames where our approach has failed (IoU less than 50) to correctly track the object. Investigating closely those cases we notice conditions where `LucidTracker` is more likely to fail. The same behaviour was observed across all three datasets. A few representatives of failure cases are visualized in Figure 10.10.

Since we are using only the annotation of the first frame for training the tracker, a clear failure case is caused by dramatic view point changes of the object from its first frame appearance, as in row 5 of Figure 10.10. The proposed approach also under-performs when recovering from occlusions: it takes several frames for the full object mask to re-appear (rows 1-2 in Figure 10.10). This is mainly due to the convnet having learnt to follow-up the previous frame mask. Augmenting the lucid dreams with plausible occlusions might help mitigate this case. Another failure case occurs when two similar looking objects cross each other, as in row 6 in Figure 10.10. Here both cues: the previous frame guidance and learnt via per-video tuning appearance, are no longer discriminative to correctly continue tracking.

We also observe that the `LucidTracker` struggles to track the fine structures or

	Method	# training images	Flow $\mathcal{F}$	DAVIS$_{16}$ J Mean ↑	J Recall ↑	J Decay ↓	F Mean ↑	F Recall ↑	F Decay ↓	T Mean ↓
	Box oracle	0	✗	45.1	39.7	**-0.7**	21.4	6.7	1.8	**1.0**
	Grabcut oracle	0	✗	67.3	76.9	1.5	65.8	77.2	2.9	34.0
Ignores 1st frame	Saliency	0	✗	32.7	22.6	-0.2	26.9	10.3	**0.9**	32.8
	NLC (Faktor and Irani, 2014)	0	✓	64.1	73.1	8.6	59.3	65.8	8.6	35.8
	MP-Net (Tokmakov *et al.*, 2017a)	~22.5k	✓	69.7	82.9	5.6	66.3	78.3	6.7	68.6
	Flow saliency	0	✓	70.7	83.2	6.7	69.7	82.9	7.9	48.2
	FusionSeg (Jain *et al.*, 2017)	~95k	✓	71.5	-	-	-	-	-	-
Uses 1st frame	Mask warping	0	✓	32.1	25.5	31.7	36.3	23.0	32.8	8.4
	FCP (Perazzi *et al.*, 2015)	0	✓	63.1	77.8	3.1	54.6	60.4	3.9	28.5
	BVS (Maerki *et al.*, 2016)	0	✗	66.5	76.4	26.0	65.6	77.4	23.6	31.6
	ObjFlow (Tsai *et al.*, 2016)	0	✓	71.1	80.0	22.7	67.9	78.0	24.0	22.1
	STV (Wang and Shen, 2017)	0	✓	73.6	-	-	72.0	-	-	-
	VPN (Jampani *et al.*, 2016a)	~2.3k	✗	*75.0*	-	-	*72.4*	-	-	*29.5*
	OSVOS (Caelles *et al.*, 2017b)	~2.3k	✗	*79.8*	*93.6*	*14.9*	*80.6*	**92.6**	*15.0*	*37.6*
	MaskTrack (Perazzi *et al.*, 2017)	~11k	✓	80.3	93.5	8.9	75.8	88.2	9.5	18.3
	LucidTracker	**24~126**	✓	**84.8**	**94.6**	4.3	**82.3**	90.5	7.0	15.8

Table 10.9: Comparison of segment tracking results on DAVIS$_{16}$ benchmark. Numbers in italic are computed based on subsets of DAVIS$_{16}$. Our LucidTracker improves over previous results.

details of the object, e.g. wheels of the bicycle or motorcycle in rows 1-2 in Figure 10.10. This is the issue of the underlying choice of the convnet architecture, due to the several pooling layers the spatial resolution is lost and hence the fine details of the object are missing. This issue can be mitigated by switching to more recent semantic labelling architectures (e.g. Pohlen *et al.* (2017); Chen *et al.* (2017a)).

Conclusion. `LucidTracker` shows robust performance across different videos. However, a few failure cases were observed due to the underlying convnet architecture, its training, or limited visibility of the object in the first frame.

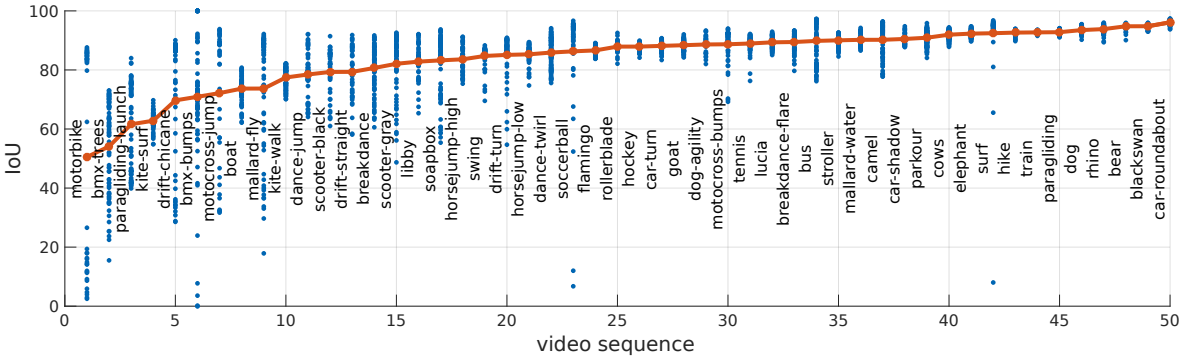


Figure 10.9: Per-sequence results on $DAVIS_{16}$.

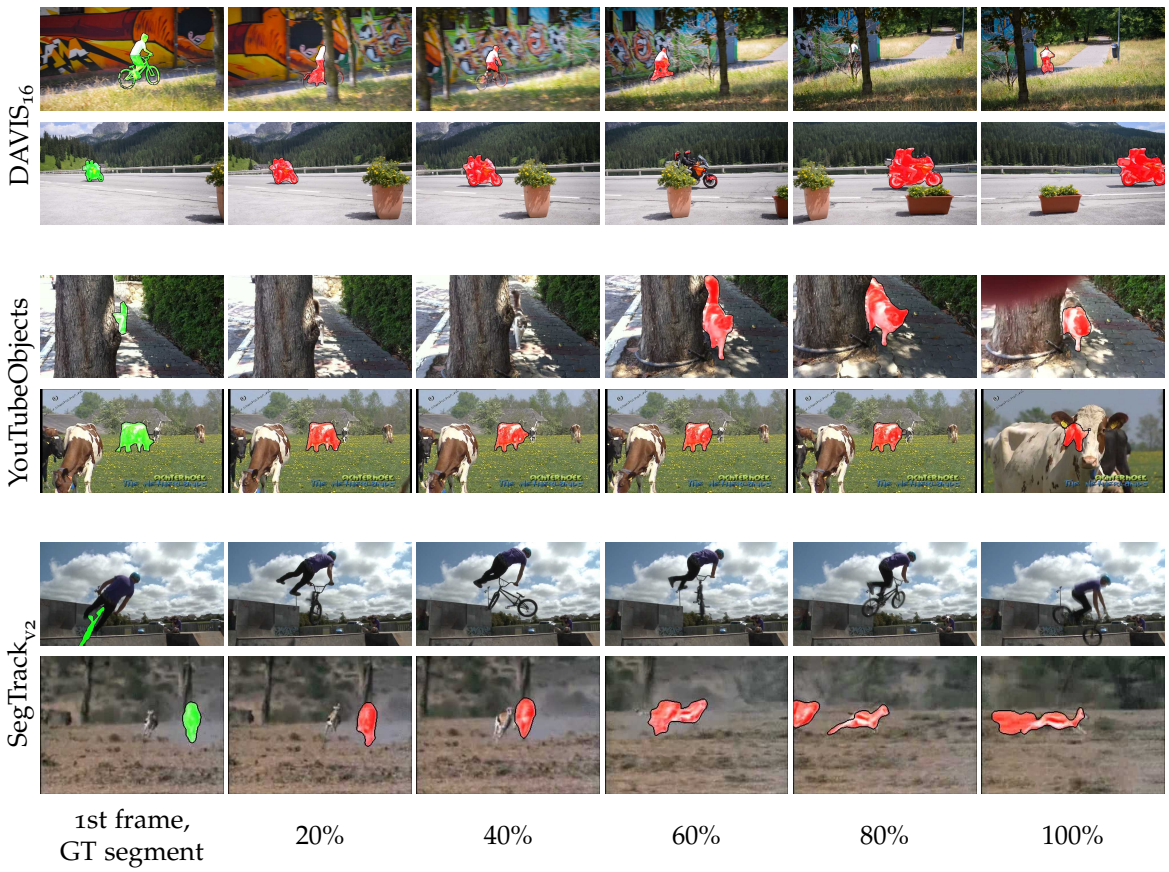


Figure 10.10: Failure cases. Frames sampled along the video duration (e.g. 50%: video middle point). For each dataset we show 2 out of 5 worst results (based on mIoU over the video).

Attribute	Method				
	BVS (Maerki *et al.*, 2016)	ObjFlow (Tsai *et al.*, 2016)	OSVOS (Caelles *et al.*, 2017b)	MaskTrack (Perazzi *et al.*, 2017)	LucidTracker
Appearance change	0.46	0.54	***0.81***	0.76	0.78
Background clutter	0.63	0.68	*0.83*	0.79	**0.85**
Camera-shake	0.62	0.72	*0.78*	0.78	**0.87**
Deformation	0.7	0.77	*0.79*	0.78	**0.87**
Dynamic background	0.6	0.67	*0.74*	0.76	**0.77**
Edge ambiguity	0.58	0.65	*0.77*	0.74	**0.78**
Fast-motion	0.53	0.55	*0.76*	0.75	**0.80**
Heterogeneous object	0.63	0.66	*0.75*	0.79	**0.83**
Interacting objects	0.63	0.68	*0.75*	0.77	**0.84**
Low resolution	0.59	0.58	***0.77***	**0.77**	0.76
Motion blur	0.58	0.6	*0.74*	0.74	**0.83**
Occlusion	0.68	0.66	*0.77*	0.77	**0.83**
Out-of-view	0.43	0.53	*0.72*	0.71	**0.84**
Scale variation	0.49	0.56	*0.74*	0.73	**0.76**
Shape complexity	0.67	0.69	*0.71*	0.75	**0.81**

Table 10.10: Attribute evaluation. LucidTracker improves across the bulk of tracking challenges.

10.6 MULTIPLE OBJECT TRACKING RESULTS

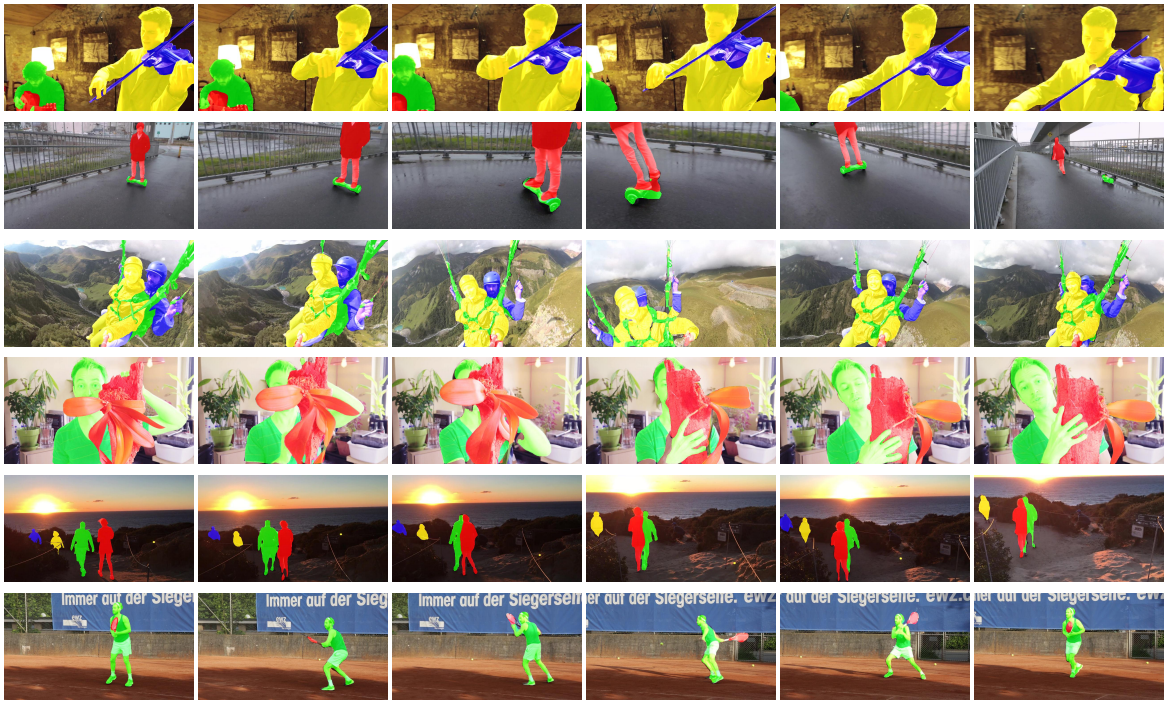
We present here an empirical evaluation of LucidTracker for multiple object tracking task: given a first frame labelled with the masks of several object instances, one aims to find the corresponding masks of objects in future frames.

10.6.1 Experimental setup

Dataset. For multiple object tracking we use the 2017 DAVIS Challenge on Video Object Segmentation[5] (Pont-Tuset *et al.*, 2017) (DAVIS_{17}). Compared to DAVIS_{16} this is a larger, more challenging dataset, where the video sequences have multiple objects in the scene. Videos that have more than one visible object in DAVIS_{16} have been re-annotated (the objects were divided by semantics) and the train and val sets were extended with more sequences. In addition, two other test sets (test-dev and test-challenge) were introduced. The complexity of the videos has increased with more distractors, occlusions, fast motion, smaller objects, and fine structures. Overall, DAVIS_{17} consists of 150 sequences, totalling 10 474 annotated frames and 384 objects.

We evaluate our method on two test sets, the test-dev and test-challenge sets,

[5]http://davischallenge.org/challenge2017



1st frame, GT segment | 20% | 40% | 60% | 80% | 100%

Figure 10.11: LucidTracker qualitative results on $DAVIS_{17}$, test-dev set. Frames sampled along the video duration (e.g. 50%: video middle point). The videos are chosen with the highest mIoU measure.

each consists of 30 video sequences, on average $\sim$ 3 objects per sequence, the length of the sequences is $\sim$ 70 frames. For both test sets only the masks on the first frames are made public, the evaluation is done via an evaluation server. Our experiments and ablation studies are done on the test-dev set.

Evaluation metric. The accuracy of multiple object tracking is evaluated using the region (J) and boundary (F) measures proposed by the organisers of the challenge. The average of J and F measures is used as overall performance score. Please refer to Pont-Tuset *et al.* (2017) for more details about the evaluation protocol.

Training details. All experiments in this section are done using the single stream architecture discussed in sections 10.3.1 and 10.5.4.3. For training the models we use SGD with mini-batches of 10 images and a fixed learning policy with initial learning rate of 10^{-3}. The momentum and weight decay are set to 0.9 and $5 \cdot 10^{-4}$, respectively. All models are initialized with weights trained for image classification on ImageNet (Simonyan and Zisserman, 2015). We then train per-video for 40k iterations.

10.6.2 Key results

Tables 10.11 and 10.12 presents the results of the 2017 DAVIS Challenge on test-dev and test-challenge sets (Pont-Tuset *et al.*). Our main results for the multi-object tracking challenge are obtained via an ensemble of four different models ($f_{\mathcal{I}}$, $f_{\mathcal{I}+\mathcal{F}}$, $f_{\mathcal{I}+\mathcal{S}}$, $f_{\mathcal{I}+\mathcal{F}+\mathcal{S}}$), see Section 10.3.1.

The proposed system, `LucidTracker`, provides the best tracking quality on the test-dev set and shows competitive performance on the test-challenge set, holding the second place in the competition. The full system is trained using the standard ImageNet pre-training initialization, Pascal VOC12 semantic annotations for the $\mathcal{S}_t$ input ($\sim 10k$ annotated images), and one annotated frame per test video, 30 frames total on each test set. As discussed in Section 10.6.3, even without $\mathcal{S}_t$ `LucidTracker` obtains competitive results (only 2 score points drop).

The top entry `lixx` (Li *et al.*, 2017b) uses a deeper convnet model (ImageNet pre-trained ResNet), a similar pixel-level tracking architecture, trains it over external segmentation data (using $\sim 120k$ pixel-level annotated images from MS-COCO and Pascal VOC for pre-training, and akin to Caelles *et al.* (2017b) fine-tuning on the DAVIS_{17} train and val sets, $\sim 10k$ annotated frames), and extends it with a box-level object detector (trained over MS-COCO and Pascal VOC, $\sim 500k$ bounding boxes) and a box-level object re-identification model trained over $\sim 60k$ box annotations (on both images and videos). We argue that our system reaches comparable results with a significantly lower amount of training data.

Figure 10.11 provides qualitative results of `LucidTracker` on the test-dev set. The video results include successful handling of multiple objects, full and partial occlusions, distractors, small objects, and out-of-view scenarios.

Conclusion. We show that top results for multiple object tracking can be achieved via our approach that focuses on exploiting as much as possible the available annotation on the first video frame, rather than relying heavily on large external training data.

10.6.3 Ablation study

Table 10.13 explores in more details how the different ingredients contribute to our results. We see that adding extra information (channels) to the system, either optical flow magnitude or semantic segmentation, or both, does provide $1 \sim 2$ percent point improvement. The results show that leveraging semantic priors and motion information provides a complementary signal to RGB image and both ingredients contribute to the tracking results.

Combining in ensemble four different models ($f_{\mathcal{I}+\mathcal{F}+\mathcal{S}} + f_{\mathcal{I}+\mathcal{F}} + f_{\mathcal{I}+\mathcal{S}} + f_{\mathcal{I}}$) allows to enhance the results even further, bringing 3 percent point gain. Our lucid dreams enable automatic CRF-tuning (see Section 10.5.3.3) which allows to further improve the results (65.2 $\rightarrow$ 66.6 mIoU).

Method	$DAVIS_{17}$, test-dev set							
	Rank	Global mean ↑	*J*			*F*		
			Mean ↑	Recall ↑	Decay ↓	Mean ↑	Recall ↑	Decay ↓
sidc	10	45.8	43.9	51.5	34.3	47.8	53.6	36.9
YXLKJ	9	49.6	46.1	49.1	22.7	53.0	56.5	22.3
haamooon (Shaban *et al.*, 2017)	8	51.3	48.8	56.9	**12.2**	53.8	61.3	**11.8**
Fromandtozh (Zhao, 2017)	7	55.2	52.4	58.4	18.1	57.9	66.1	20.0
ilanv (Sharir *et al.*, 2017)	6	55.8	51.9	55.7	17.6	59.8	65.8	18.9
voigtlaender (Voigtlaender and Leibe, 2017a)	5	56.5	53.4	57.8	19.9	59.6	65.4	19.0
lalalafine123	4	57.4	54.5	61.3	24.4	60.2	68.8	24.6
wangzhe	3	57.7	55.6	63.2	31.7	59.8	66.7	37.1
lixx (Li *et al.*, 2017b)	2	66.1	**64.4**	73.5	24.5	67.8	75.6	27.1
LucidTracker	1	**66.6**	63.4	**73.9**	19.5	**69.9**	**80.1**	19.4

Table 10.11: Comparison of segment tracking results on $DAVIS_{17}$, test-dev set. Our LucidTracker shows top performance.

Method	$DAVIS_{17}$, test-challenge set							
	Rank	Global mean ↑	*J*			*F*		
			Mean ↑	Recall ↑	Decay ↓	Mean ↑	Recall ↑	Decay ↓
zwrq0	10	53.6	50.5	54.9	28.0	56.7	63.5	30.4
Fromandtozh (Zhao, 2017)	9	53.9	50.7	54.9	32.5	57.1	63.2	33.7
wasidennis	8	54.8	51.6	56.3	26.8	57.9	64.8	28.8
YXLKJ	7	55.8	53.8	60.1	37.7	57.8	62.1	42.9
cjc (Cheng *et al.*, 2017)	6	56.9	53.6	59.5	25.3	60.2	67.9	27.6
lalalafine123	6	56.9	54.8	60.7	34.4	59.1	66.7	36.1
voigtlaender (Voigtlaender and Leibe, 2017a)	5	57.7	54.8	60.8	31.0	60.5	67.2	34.7
haamooon (Shaban *et al.*, 2017)	4	61.5	59.8	71.0	21.9	63.2	74.6	23.7
vantam299 (Le *et al.*, 2017)	3	63.8	61.5	68.6	**17.1**	66.2	79.0	**17.6**
LucidTracker	2	67.8	65.1	72.5	27.7	70.6	**79.8**	30.2
lixx (Li *et al.*, 2017b)	1	**69.9**	**67.9**	**74.6**	22.5	**71.9**	79.1	24.1

Table 10.12: Comparison of segment tracking results on $DAVIS_{17}$, test-challenge set. Our LucidTracker shows competitive performance, holding the second place in the competition.

Variant	$\mathcal{I}$	$\mathcal{F}$	$\mathcal{S}$	ensemble	CRF tuning	DAVIS$_{17}$ test-dev global mean	DAVIS$_{17}$ test-dev mIoU	DAVIS$_{17}$ test-dev mF	DAVIS$_{17}$ test-challenge global mean	DAVIS$_{17}$ test-challenge mIoU	DAVIS$_{17}$ test-challenge mF
LucidTracker (ensemble)	✓	✓	✓	✓	✓	**66.6**	**63.4**	**69.9**	**67.8**	**65.1**	**70.6**
	✓	✓	✓	✓	✗	65.2	61.5	69.0	-	-	-
	✓	✓	✗	✓	✓	64.9	61.3	68.4	-	-	-
	✓	✓	✗	✓	✗	64.2	60.1	68.3	-	-	-
LucidTracker $\mathcal{I}+\mathcal{F}+\mathcal{S}$	✓	✓	✓	✗	✓	62.9	59.1	66.6	-	-	-
	✓	✓	✓	✗	✗	62.0	57.7	62.2	64.0	60.7	67.3
$\mathcal{I}+\mathcal{F}$	✓	✓	✗	✗	✗	61.3	56.8	65.8	-	-	-
$\mathcal{I}+\mathcal{S}$	✓	✗	✓	✗	✗	61.1	56.9	65.3	-	-	-
$\mathcal{I}$	✓	✗	✗	✗	✗	59.8	63.1	63.9	-	-	-

Table 10.13: Ablation study of different ingredients. DAVIS$_{17}$, test-dev and test challenge sets.

Conclusion. The results show that both flow and semantic priors provide a complementary signal to RGB image only. Despite its simplicity our ensemble strategy provides additional gain and leads to competitive results. Notice that even without the semantic segmentation signal $\mathcal{S}_t$ our ensemble result is competitive.

10.6.4 Error analysis

We present the per-sequence results of `LucidTracker` on DAVIS$_{17}$ in Figure 10.12 (per frame results not available from evaluation server). We observe that this dataset is significantly more challenging than DAVIS$_{16}$ (compare to Figure 10.9), with only $1/3$ of the test videos above 80 mIoU. This shows that multiple object tracking is a much more challenging task than tracking a single object.

The failure cases discussed in Section 10.5.5 still apply to the multiple objects case. Additionally, on DAVIS$_{17}$ we observe a clear failure case when tracking similar looking object instances, where the object appearance is not discriminative to correctly track the object, resulting in label switches or bleeding of the label to other look-alike objects. Figure 10.13 illustrates this case. This issue could be mitigated by using object level instance identification modules, like Li *et al.* (2017b), or by changing the training loss of the model to more severely penalize identity switches.

Conclusion. Albeit the `LucidTracker` results remain robust across different videos, overall results are lower than for the single object tracking case showing that there is more room for improvement in the multiple object pixel-level tracking task.

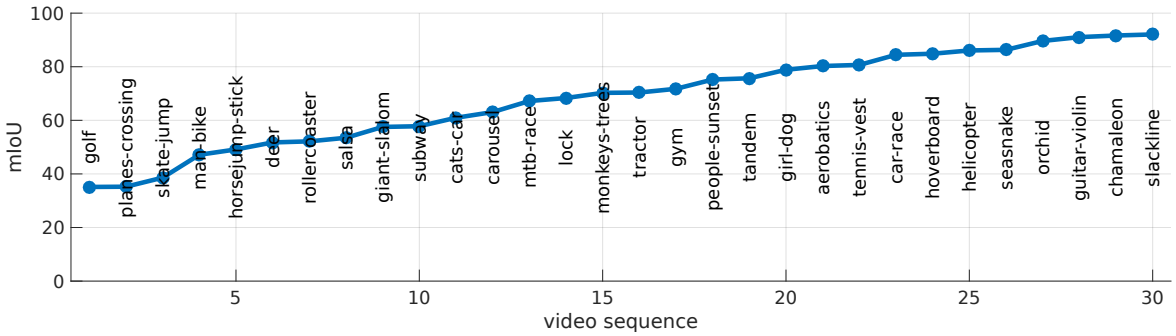


Figure 10.12: Per-sequence results on $DAVIS_{17}$, test-dev set.

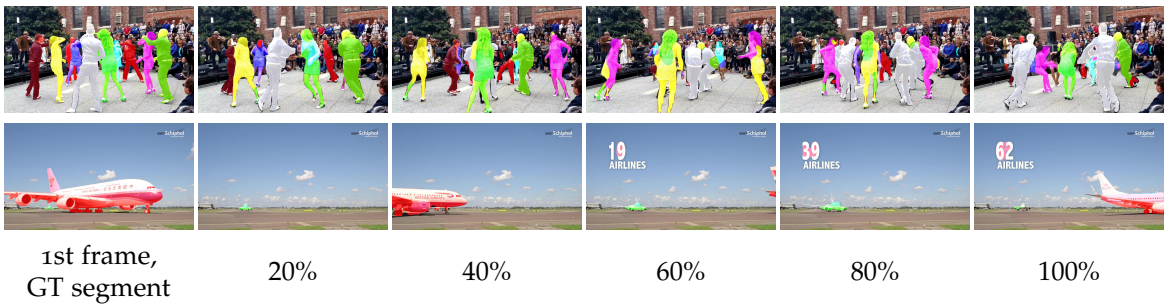


Figure 10.13: LucidTracker failure cases on $DAVIS_{17}$, test-dev set. Frames sampled along the video duration (e.g. 50%: video middle point). We show 2 results mIoU over the video below 50.

10.7 CONCLUSION

We have described a new convnet-based approach for pixel-level object tracking in videos. In contrast to previous work in Chapter 9, we show that top results in single and multiple object tracking can be achieved without requiring external training datasets (neither images with saliency annotation nor annotated videos). Even more, our experiments indicate that it is not always beneficial to use additional training data, synthesizing training samples close to the test domain is more effective than adding more training samples from related domains.

Our extensive analysis decomposed the ingredients that contribute to our improved results, indicating that our new training strategy and the way we leverage additional cues such as semantic and motion priors are key.

Showing that training a convnet for object tracking can be done with only few (~ 100) training samples changes the mindset regarding how much general "objectness" knowledge is required to approach this problem (Perazzi *et al.*, 2017; Jain *et al.*, 2017; Voigtlaender and Leibe, 2017b), and more broadly how much training data is required to train large convnets depending on the task at hand.

We hope these new results will fuel the ongoing evolution of convnet techniques for single and multiple object tracking.

CONCLUSIONS AND FUTURE PERSPECTIVES 11

Significant progress has been achieved in image and video segmentation over the last years (Long *et al.*, 2015; Bansal *et al.*, 2017; He *et al.*, 2017; Tokmakov *et al.*, 2017b; Caelles *et al.*, 2017b). To a large extent the success of the current methods can be attributed to the strong appearance models completely learned from data, in particular using deep convolutional neural networks. As the complexity of methods increases and so the number of model parameters that have to be estimated from data, large representative training sets are crucial for best performance. For instance, the common practice is to pre-train the convnet on $\sim 10^6$ training samples for ImageNet classification (Russakovsky *et al.*, 2015)), and then in order to shift from the pre-training domain to the application domain by fine-tuning on thousands of images with pixel-level annotations (Everingham *et al.*, 2015; Lin *et al.*, 2014; Pont-Tuset *et al.*, 2017). However, the large-scale segmentation datasets which enable these high-quality results require a lot of human labor to be annotated and sometimes prohibitive to obtain, e.g. in case of video segmentation. This considerably restricts the potential to transfer these models to approach different domains or various object categories. As a consequence, methods with alternative weaker forms of supervision (Jain *et al.*, 2017; Papandreou *et al.*, 2015; Kolesnikov and Lampert, 2016b) or synthetic data (Mayer *et al.*, 2016; Tokmakov *et al.*, 2017a) for training have received a lot of attention recently as well as unsupervised and self-supervised techniques (Li *et al.*, 2015; Sermanet *et al.*, 2017; Xiao and Lee, 2016).

In this thesis we have looked at three research directions, which we briefly summarize in the following. In the first direction, *image segmentation with weaker forms of supervision*, we focused on training convolutional networks with bounding box or image label supervision for object boundary and semantic/instance labelling tasks. We proposed approaches to generate pixel-level approximate groundtruth from these weaker forms of annotations to train a network, which allows to achieve high-quality results without any modifications of the architecture or the training procedure. We also contributed with the investigated recursive training of convnets for weakly supervised image labelling and to the best of our knowledge we were the first to address the problems of weakly supervised object boundaries and instance segmentation. In the second direction, *graph-based video segmentation*, we moved from images to learning to segment in videos. We addressed the problem of the excessive computational and memory costs inherent to solving video segmentation via graphs and contributed with learning a better and more efficient representation of the graph from the available training data. In the third direction, *pixel-level object tracking via CNNs*, we considered the task of propagating the mask of an object throughout a video given its annotation in the first frame. We addressed the challenge of the limited amount of densely annotated consequent video data

for training by introducing a way to train the network from static images only and generating in-domain synthetic data from the given first frame mask. In all three directions we have advanced the state of the art on multiple challenging benchmarks. Furthermore, we contributed to the field by making the source code, trained models and generated data for training freely available to the community.

In this chapter we further discuss and detail the contributions of the thesis (Section 11.1) and then review open problems and potential future perspectives (Section 11.2).

11.1 DISCUSSION OF CONTRIBUTIONS

The overall goal of this thesis was to exploit learning to segment in images and videos using different levels of supervision during training and test time. We tackled three specific sub-topics, namely *image segmentation with weaker forms of supervision*, *video segmentation via graphs* and *pixel-level object tracking via CNNs*, as introduced earlier. In the following we will discuss the main results and insights that this thesis contributes with respect to the individual chapters.

11.1.1 Image segmentation with weaker forms of supervision

In the first two chapters we explore how to train the model using bounding box supervision. In Chapter 3 we introduced the problem of weakly supervised object boundary detection. We mainly focused on external object boundaries, which can be seen as contours of object mask, as well as class-specific (semantic) object boundaries. For the experiments we considered using two types of the boundary detectors: a decision forest (Dollár and Zitnick, 2015) and a convnet-based edge detector (Xie and Tu, 2015). We demonstrated that noisy boundaries generated from unsupervised methods (Canny, 1986; Felzenszwalb and Huttenlocher., 2004) can be a source for supervision for learning based detectors and that training methods are quite robust to the noise in the generated annotations. Since accurate boundaries tend to have consistent appearance, while erroneous detections are mostly inconsistent they are able to pick-up the correct signal. Based on this insight we proposed to generate pixel-level approximate groundtruth for object boundary detection using the supervision from a bounding box object detector (Girshick, 2015; Ren *et al.*, 2015). For generating boundary annotations we fused unsupervised image segmentation (Felzenszwalb and Huttenlocher., 2004), GrabCut (Rother *et al.*, 2004) and object proposal methods (Uijlings *et al.*, 2013; Pont-Tuset *et al.*, 2016) using the consensus strategy to de-noise the predictions. We showed that this approximate groundtruth alone suffice to train a boundary detector and as a result to achieve high-quality object boundary estimates. We reported state-of-the-art performance on the Pascal VOC12 (Everingham *et al.*, 2015) and SBD (Hariharan *et al.*, 2011) benchmarks. With the proposed weak supervision technique we achieved the top performance for object and class-specific boundaries, outperforming by a large margin previously

reported fully supervised results. We demonstrated that while training bounding box detector one could have the object boundary detector for free, without any additional annotation effort.

In Chapter 4 we extend the proposed in Chapter 3 approach to other closely related tasks, such as semantic labelling and instance segmentation. We contributed with providing new insights on how to train pixel-labelling convnets from bounding boxes only. We observed that when applying the model over the training set, the network outputs capture the object shape significantly better than just boxes. This motivated us to explore recursive training as a de-noising strategy of box annotations, where convnet predictions of the previous training round are used as supervision for the next round. We showed that when carefully employing the available cues given by the shape of the box and object priors, recursive training with rectangles as input can be surprisingly effective. Even more, generated training labels via classic techniques for box-guided instance segmentation (Rother *et al.*, 2004; Pont-Tuset *et al.*, 2016) can serve as good supervision for pixel-labelling convnet. When finding a good balance between accuracy and recall in the noisy training segments, a single training round is enough to get high-quality results. With the proposed technique we improved over previously reported weakly supervised results for semantic segmentation on Pascal VOC12 (Everingham *et al.*, 2015) and reached $\sim$95% of the quality of the same network trained on the ground truth segmentation annotations over the same data. By employing extra training data with bounding box annotations from COCO (Lin *et al.*, 2014) we matched the full supervision results. Furthermore, we were the first to show that similar results for weakly supervised instance segmentation.

In Chapter 5 we moved to a weaker form of supervision to train a semantic segmentation convnet - image label annotations. Training a convnet with image-level supervision is a much harder task compared to box supervision, as image labels can only provide a constraint of the presence of the class. Therefore, one of the main challenges is outputting the full extent of the object. To deal with this issue we decomposed the problem into two: finding the object location and finding it's extent. We used high confidence points of the activation maps of the trained classifiers (seeds) to locate the object (Zhou *et al.*, 2016). We analyzed different factors that influence the seeds generation. Our experiments showed that better classifiers do not automatically make better seeders and higher resolution networks are usually better localisers. For finding the object extent we employed a weakly-supervised class-agnostic saliency model, exploiting the assumption that a large number of photos aim at capturing a subject and thus finding the object extent can be seen as finding the background area in an image. We then combined cues from the seeds and saliency via the proposed "guide labeller" to produce a rough segmentation annotation for training a convnet in a regular fully-supervised fashion. With this approach we were able to recover 80% of the fully supervised performance, which was the new state of the art in semantic labelling with image label supervision.

11.1.2 Video segmentation via graphs

In the next three chapters we focused on learning to segment in videos using graphs. We addressed one of the important limitation of graph-based methods - significant computational and memory costs - by learning from the available training data how to construct a smaller, sparser and better graph. Our contributions allowed to improve the scalability of graph-based approaches and made their use possible for today's high quality video data.

In Chapter 6 we showed that graph-based video segmentation methods could greatly benefit from the integration of information learned from training data in terms of must-link constraints. We formalized the use of learned must-link constraints in conjunction with spectral techniques and proposed the relevant learning and inference algorithms. Experimentally we demonstrated on two different benchmarks that learned must-link constraints improve the performance by guiding spectral clustering towards a desired segmentation and reduce the required runtime and memory footprint by building a graph on coarser superpixels merged based on learned must-link constraints.

In Chapter 7 we focused how to construct a graph in order to improve video segmentation performance as well as to reduce the problem size without changing the graph partitioning method. We proposed an empirical approach to learn both the edge topology and weights of the graph. We combined well-established features by means of a random forest classifier and learnt to calibrate the classifier output scores by its accuracy. In addition, we altered the graph topology by selecting the most confident edges. Our method of learning the graph improved the results of the best performing video segmentation algorithm by 6% on the challenging VSB100 benchmark (Galasso *et al.*, 2013), while reducing its runtime by 55%, as the learnt graph is much sparser.

In Chapter 8 we improved superpixels - the graph nodes themselves - which are the starting point for estimating pairwise terms, and thus directly influence the final quality of graph-based video segmentation techniques. We provided the comparative evaluation of existing superpixel/voxel methods, indicating the importance of the initial superpixels/voxels for graph-based video segmentations. Our finding was that classical superpixel/voxel methods (Chang *et al.*, 2013; Achanta *et al.*, 2012; Bergh *et al.*, 2013) underperform and boundary based superpixels, extracted via hierarchical image segmentation, are more effective for the task. Based on this insight, we proposed an approach to improve boundary estimates, and therefore superpixels, specifically for videos. We fused image and time domain cues as well as integrated high-level object-related cues into the local image segmentation processing. With this technique we significantly enhanced boundary estimation in video frames and as a result obtained improved per-frame superpixels. When using superpixels built over these improved boundaries, we observed consistent improvement over two different video segmentation methods (Galasso *et al.*, 2013, 2014) and two different datasets. Our analysis of the results indicated that the improvement was at most in the cases where baseline methods degraded.

11.1.3 Pixel-level object tracking via CNNs

In the last two chapters we focused on pixel-level object tracking via convolutional networks and addressed the inherent challenges of limited densely annotated video data for training and the problem of domain shift.

In Chapter 9 we presented MaskTrack, a novel convnet-based approach to video object segmentation that uses only static images for training instead of relying on consequent video data. We employed a pixel labelling convnet to process video sequence per-frame, using the output of the previous frame as an additional input channel, that served as a guidance towards the object of interest in the next frame. In this way the network was designed to consider as input only one frame and the rough binary mask (the previous frame estimate), which could be easily synthesized from the ground truth mask via affine and non-rigid transformations. This enabled to train the network with existing large-scale image datasets and avoid using expensive densely annotated video data for training. The proposed system reached state-of-the-art performance on three extremely heterogeneous video segmentation benchmarks, using the same model and parameters across all videos in contrast to previous work. We provided a detailed ablation study of different ingredients of the model. The key component of the proposed approach was online fine-tuning of the network on the given first frame annotation of the test video, which allowed to capture the appearance of the specific object instance and thus made the performance more robust to challenging situations inherent in video data, such as occlusions and fast motion. We showed that our method could handle different types of input annotations and our results were competitive even when using only bounding box annotations, instead of segmentation masks. In addition, we explored the effect of varying the amount of annotated frames per video during online fine-tuning. We demonstrated that with only one annotation every 10th frame we can reach 85% mIoU quality. This makes the proposed system suitable for diverse applications with different requirements in terms of accuracy and efficiency.

In Chapter 10 we extended the approach proposed in the previous chapter with better integration of motion cues as well as semantic information, making the gains across different datasets more stable. For this we altered the network architecture to accept optical flow magnitude and semantic priors as additional input channels. Combining the appearance with motion and semantic cues enabled the model to segment better and improved the temporal coherency. We also relaxed the dependence of using $\sim 10k$ pixel-level image annotations for training, as in Chapter 9, by introducing Lucid Data Dreaming, an automated approach to synthesize training data for pixel-level object tracking. We proposed to generate multiple plausible future frames using the given first frame image and its mask of the test video, ensuring a sufficient amount of training samples close to the test domain. Employing the lucid dream images for training enabled to achieve the top results while using only ~ 100 individual annotated training frames, which was $20\times \sim 100\times$ less than previous approaches (Caelles *et al.*, 2017b; Voigtlaender and Leibe, 2017b). We conducted an extensive analysis to explore the factors contributing to our results. Our experiments

indicated that it is not always beneficial to employ additional external training data and using few training frames close to the test domain is more effective than using larger training volumes across domains. Furthermore, we explored training the pixel-level object tracking network with only a single annotated frame and zero pre-training. With such minimal amount of training data we obtained competitive performance, demonstrating the effectiveness of Lucid Data Dreaming and changing the mindset how much training data is required for the object tracking task. We showed that our approach is suitable for both single and multiple object tracking, taking the second place in the DAVIS Challenge[6] (Pont-Tuset *et al.*, 2017).

11.2 FUTURE PERSPECTIVES

In this section we first discuss limitations of the presented work as well as potential next steps towards image and video segmentation and speculate about promising research directions. Then, we conclude this section with giving a broader view on the topic in Section 11.2.3.

11.2.1 Image segmentation

In the thesis we have discussed several challenges towards image segmentation with different levels of supervision and provided possible solutions to address them (Chapters 3 – 5). However, our research in this thesis leads to some open issues that we would like to discuss in the following.

Exploring different convnet architectures. Improving the boundary adherence of the convnet predictions as well as segmenting objects at multiple scales is still an open problem for image segmentation. Several solutions have been proposed recently that showed promising results, such as different variants of spatial pyramid pooling (Zhao *et al.*, 2016; Chen *et al.*, 2016b) or exploiting image-level features for global context information (Liu *et al.*, 2015; Chen *et al.*, 2017a). We believe that combining multi-scale information with global context is a promising direction. However, so far very little analysis has been performed to showcase the advantages and limitations of these different architecture alternations. Comparing the behavior of these network variants in terms of different appearance factors would help to understand how to better represent and handle visual information at different scales, which is crucial for advancing the state of the art.

Data synthesis. Although deep learning methods are quite powerful, they are dependent on data to be able to learn the necessary representation. Therefore, a straightforward way to boost the performance is to provide more data for training. However, obtaining pixel-level annotations is tedious and expensive. One of the alternatives is to generate synthetic data by recomposing real world images, su-

[6] http://davischallenge.org/challenge2017

perimposing the objects into different scenes. This provides more realistic results and has better generalization qualities compared to renderings of 3D models. Using synthetically generated composite images for training has shown promise for object detection (Debidatta Dwibedi, 2017; Georgakis *et al.*, 2017) and human pose estimation (Park and Ramanan, 2015), but yet to be shown for semantic and instance segmentation. Special care should be taken of differences in lighting conditions, scaling, blending of boundaries and selection of the object position in a new scene and their impact on the performance of the learned models. Another and more appealing option is to generate more training data by doing labels-to-image translation via Generative Adversarial Networks (GANs), which recently showed high quality results (Zhu *et al.*, 2017a), comparable to natural images. Besides, multiple new scenes can be generated by recomposing label maps, which is a much easier task than synthesizing realistic looking images via classic computer graphics tools. We believe that the proposed strategies can further boost the results and help to relax the data dependency constraint.

Exploiting unlabeled/weakly-labeled data. A huge amount of unlabeled/weakly-labeled data with diverse context is being generated every second around the world and in many cases becomes immediately available on the Internet. New deep learning algorithms should be designed to take advantage of such data. Recently a few works have explored using web-crawled images and videos with class label annotations as additional training data for semantic segmentation convnets (Jin *et al.*, 2017; Hong *et al.*, 2017). The main challenge in this line of research is de-noising the data crawled from the Internet. Another line of work proposed to learn a visual representation using only the innate structure of images and videos as a source of supervision (Doersch *et al.*, 2015; Goyal *et al.*, 2017; Wang and Gupta, 2015; Zhang *et al.*, 2017). Although these approaches showed promising results, so far they could not match the full supervision results. There is still much to explore and to enhance. One interesting direction might be to exploit temporal coherence and dynamics in videos as supervision since video data contains richer information than static images.

Domain adaptation. Synthesizing or web-crawling additional training data can lead to statistical deviations from the target domain. While the differences between domains might appear mild to a human, it can make the benefits of training with additional data non-existent and in the extreme case result in the much lower performance. The problem of domain shift could be mitigated with domain adaptation techniques. In consequence of ever growing demand for more data of deep learning methods this topic has received a lot of attention recently (Ganin and Lempitsky, 2015; Long *et al.*, 2016; Bousmalis *et al.*, 2017; Tzeng *et al.*, 2017). We see the problem of domain adaptation as one of the main challenges for exploiting synthetic and weakly-labeled data to boost convnets performance for image segmentation and other related vision tasks.

Integrating higher-level information. Image segmentation can be improved by incorporating high-level concepts. However, while semantic reasoning is natural for humans, it is not trivial for machines. Specifically, integrating external knowledge into deep learning methods is challenging. One possible direction would be to leverage the structure of objects in order to constrain better convnet predictions. For instance, the model can learn that a car has four wheels, a bumper and a hood, while a motorcycle has two wheels and a fork that connects a front wheel to the frame. This knowledge of the object structure would help to better recover the object mask and decrease the misclassification error. Another approach would be to integrate the global scene aspects, i.e. how natural and plausible is the predicted scene, in order to avoid occurrence of classes in improbable situations, e.g. it is very unlikely to see zebra in the urban scene or a car flying in the air. A few early works took steps towards these directions (Gould *et al.*, 2009; Socher *et al.*, 2011); however, they considered a restricted semantic setting. Future works should look into providing richer external knowledge to the networks.

11.2.2 Video segmentation

We have discussed different aspects of video segmentation via graphs (Chapters 6 – 8) and CNNs (Chapters 9, 10) in this thesis. The challenges and next steps related to learning to segment in images, discussed previously, are also inherent to video segmentation. In the following we review the possible extensions of the approaches proposed in Chapters 9, 10, open problems for addressing video segmentation with deep learning and future research directions.

Online adaptation. Since the pixel-level object tracking convnets in Chapters 9, 10 uses only the first frame annotation for per-video fine-tuning, they sometimes have troubles to adapt to drastic changes in object appearance or camera viewpoint, causing the loss of the object or the drift of the mask. One of the possible solutions to this problem is online adaption of the network to the future frames (Nam and Han, 2016; Ellis and Zografos, 2013; Bai *et al.*, 2010). Recently Voigtlaender and Leibe (2017b) proposed to update the convnet online on each frame using training examples selected based on the confidence of the network and the spatial distance. However, relying on the network predictions for online adaptation can result in error propagation and cause identity-switches when tracking multiple interacting objects. One way to resolve this issue it to integrate instance level semantic information (Dai *et al.*, 2016a; He *et al.*, 2017) and in addition to rely on temporal consistency (e.g. using forward and backward flow (Ilg *et al.*, 2017) to propagate labels) to de-noise the training examples for online tuning.

Improving data synthesis for object tracking. In Chapter 10 we introduced the lucid data dreaming synthesis scheme which has proven to be successful for pixel-level object tracking. However, there are several possible ways to enhance it. The drawbacks of the proposed approach are a very naive modeling of the foreground

and background motion and random placement of the object while recomposing the scene. We believe that incorporating semantic knowledge of the object and the scene into the simulation of object and camera motion as well as the new scene compositing would help to produce more realistic images. Furthermore, more care should also be taken of modeling changes in lighting conditions and blending of object boundaries.

Weaker forms of supervision in the first frame. One of the main disadvantages of semi-supervised video object segmentation is using an expensive object mask supervision at test time. In Chapters 4 and 5 we have shown that an object mask can be recovered with weaker forms of supervision, such as bounding box or object class label. A natural extension would be to integrate these approaches with the method proposed in Chapter 10. Another possible way obtaining the object mask is to use language descriptions of the object. It is much easier for the user to say: "I want a person on the right in a white t-shirt to be tracked", than provide a pixel-level annotation, which is a tedious task. Recently, it has been shown that textual phrases can be grounded (localized) in images and videos using an attention mechanism (Ramanishka *et al.*, 2017; Rohrbach *et al.*, 2016). We consider combining language grounding with video segmentation an interesting research direction.

Exploiting long-range temporal context. Methods proposed in Chapters 9, 10 propagate the mask only across neighbouring frames relying on temporal continuity. However, real-life videos may exhibit severe deformations and occlusions. As a consequence using only the information from the previous frame can lead to difficulties handling large displacement of objects and loss of the object. To cope with inter-object occlusions and pose variations in dynamic scenes, Li *et al.* (2017b) have recently proposed to employ object re-identification to retrieve instances that are missing during the mask propagation process. We believe that exploiting long-range temporal context of the video data is the key to obtaining robust and globally consistent segmentation.

Encoding temporal dimension. Incorporating temporal information inherent in video data into deep architectures is challenging and not straightforward. Some works have proposed to use different variants of recurrent neural networks (Siam *et al.*; Tokmakov *et al.*, 2017b,b). However, these methods suffer from the lack of large-scale densely annotated video datasets and extensive computational demands. The other group of approaches has successfully employed spatio-temporal 3D convolutions for action recognition and scene classification (Varol *et al.*, 2016; Tran *et al.*, 2015). Though, employing the fixed-sized spatio-temporal receptive fields might be not suitable for dense predictions as association of the pixels in temporal dimension is different from the spatial due to the large displacements in the dynamic scenes.

Extending semantic and instance image segmentation to videos. Many advances have been made in semantic and instance segmentation of static images (e.g. (Chen

et al., 2016b; Pohlen *et al.*, 2017; Dai *et al.*, 2016c; He *et al.*, 2017)). However, so far extending these tasks to video data has received less attention (Liu and He, 2015; Kundu *et al.*, 2016; Nilsson and Sminchisescu, 2016), especially in the context of deep learning. One of the obvious reasons is the lack of a large volume of annotated training data. Recently, a new benchmark (Richter *et al.*, 2017) has been released, which provides ground truth for low-level as well as higher-level tasks, including semantic instance segmentation and tracking, for $\sim 250k$ frames. Availability of the large-scale training data creates new opportunities for progress and development of deep network architectures that leverage the temporal structure of video.

Convnets on graphs. In Chapters 6–8 and Chapters 9, 10 we discussed approaching video segmentation via "old school" graphs or by using popular convolutional networks respectively. An interesting future research direction is to combine the best of both worlds. Recently, variants of neural networks which operate on graphs have been introduced (Kipf and Welling, 2017; Defferrard *et al.*, 2016; Li *et al.*, 2016c; Manessi *et al.*, 2017), including applications to computer vision tasks (Liang *et al.*, 2016; Li *et al.*, 2017a). Research on this topic is just getting started. Exciting developments have been made, but it remains to be seen how neural networks on graphs can be further tailored to specific types of problems, including image and video segmentation.

11.2.3 A broader outlook

While in the previous sections we have discussed concrete ideas to approach limitations and future steps with respect to the contributions of this thesis, in this section we outline a broader view on the topic.

More labelled data and effort in its sharing. Over the last few years, significant progress has been made in the field of static image understanding, in particular image segmentation. Most of the advances have come with the creation of large-scale datasets, such as ImageNet (Russakovsky *et al.*, 2015), Pascal VOC (Everingham *et al.*, 2015) and COCO (Lin *et al.*, 2014). However, when it comes to video segmentation we are still struggling to figure out how to encode the whole video volume and what are the most promising directions to move forward. One of the reasons for this is the absence of large diverse datasets for video segmentation. A few video benchmarks have been released recently (Richter *et al.*, 2017; Pont-Tuset *et al.*, 2017), which have helped and will help to advance as well as reveal the shortcomings of existing approaches. Still, in terms of video sequence number those datasets do not match in scale the existing image datasets. Therefore, collaborations in data sharing among the research groups and the industry companies, which have much more resources at hand, are strongly encouraged. Availability of annotated video data for training and evaluation will accelerate the development of this highly dynamic research area.

Bringing in multiple modalities. Enabled by the recent development and accessibility of data acquisition hardware, multi-modal data, which contains the observation from multiple modalities such as image, audio, motion, depth, etc., has become omnipresent. Capturing appearance of the visual world into a 2D plane is not enough to describe the complex environment that surrounds us. Combining data from multiple sources can not only lead to additional observations, but also result in complementary information across modalities. This gives an opportunity for computer vision systems to better understand real-word scenes, e.g. extending monocular system with stereo-vision or depth-perception sensors provides information of the 3D structure of the scene. Hence, in the real-world applications the segmentation approaches should be able to make the best use of easier access to multi-modal data, by combining it as well as employing it to resolve ambiguities caused by partial observations of a specific modality.

Learning in the wild. The challenge of putting computer vision systems into real-life settings is that the environments are constantly changing. Deep learning systems are biased towards the data they were trained with and the task they were trained for. Therefore these approaches would have troubles generalizing to new scenes, objects or tasks, even if they are very similar to the ones that they were originally trained on/for. Besides these systems do not employ any high-level processes such as conceptual abstraction or causal reasoning. In contrast, humans have no troubles adapting to new environments by learning through interaction with external world and as a result sensorimotor activity (Smith and Gasser, 2005). New computer vision systems should not entirely rely on ever growing labelled amount of training data and explore other paradigms, mimicking the way humans learn. We believe that future works should move from the controlled environments to the open sets, enabling the system to learn in the wild through trial and error as well as integrating higher-level knowledge and reasoning. In this context, reinforcement learning, self-supervised learning and recurrent neural networks with memory mechanisms seem like a promising research directions.

LIST OF FIGURES

LIST OF TABLES

BIBLIOGRAPHY

R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Suesstrunk (2012). SLIC Superpixels Compared to State-of-the-art Superpixel Methods, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 125, 127, 130, and 192.

A. Alexandrescu and K. Kirchhoff (2007). Data-Driven Graph Construction for Semi-Supervised Graph-Based Learning in NLP, in *Conf. of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies 2007*. Cited on page 111.

S. Alpert, M. Galun, A. Brandt, and R. Basri (2012). Image Segmentation by Probabilistic Bottom-Up Aggregation and Cue Integration, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 99.

B. Andres, J. H. Kappes, T. Beier, U. Köthe, and F. A. Hamprecht (2011). Probabilistic Image Segmentation with Closedness Constraints, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2011*. Cited on pages 97 and 111.

P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik (2011). Contour Detection and Hierarchical Image Segmentation, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 4, 15, 16, 35, 37, 38, 39, 40, 59, 67, 97, 99, 100, 101, 103, 105, 109, 111, 112, 113, 115, 126, 127, 130, 131, 132, 135, and 136.

A. Arnab and P. H. S. Torr (2016). Bottom-up Instance Segmentation using Deep Higher-Order CRFs, in *Proc. of the British Machine Vision Conf. (BMVC) 2016*. Cited on page 22.

Ç. Aytekin, E. C. Ozan, S. Kiranyaz, and M. Gabbouj (2015). Visual saliency by extended quantum cuts, in *Proc. IEEE International Conf. on Image Processing (ICIP) 2015*. Cited on page 172.

V. Badrinarayanan, I. Budvytis, and R. Cipolla (2013). Mixture of Trees Probabilistic Graphical Model for Video Segmentation, *International Journal of Computer Vision (IJCV)*. Cited on page 127.

V. Badrinarayanan, F. Galasso, and R. Cipolla (2010). Label propagation in video sequences, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2010*. Cited on pages 27 and 28.

V. Badrinarayanan, A. Kendall, and R. Cipolla (2015). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, *arXiv*, vol. abs/1511.00561. Cited on page 18.

M. Bai and R. Urtasun (2017). Deep Watershed Transform for Instance Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on page 22.

X. Bai, J. Wang, and G. Sapiro (2010). Dynamic Color Flow: A Motion-Adaptive Color Model for Object Segmentation in Video, in *Proc. of the European Conf. on Computer Vision (ECCV) 2010*. Cited on page 196.

X. Bai, J. Wang, D. Simons, and G. Sapiro (2009). Video snapcut: robust video object cutout using localized classifiers, in *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH) 2009*. Cited on page 31.

C. Bailer, B. Taetz, and D. Stricker (2015). Flow Fields: Dense Correspondence Fields for Highly Accurate Large Displacement Optical Flow Estimation, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on page 147.

D. Banica, A. Agape, A. Ion, and C. Sminchisescu (2013). Video Object Segmentation by Salient Segment Chain Composition, in *ICCV, IPGM Workshop 2013*. Cited on pages 26, 97, and 109.

D. Banica and C. Sminchisescu (2015). Second-order constrained parametric proposals and sequential search-based structured prediction for semantic segmentation in RGB-D images, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 35.

A. Bansal, X. Chen, B. Russell, A. Gupta, and D. Ramanan (2017). PixelNet: Representation of the pixels, by the pixels, and for the pixels, *arXiv:1702.06506*. Cited on pages 27 and 189.

J. Barron and B. Poole (2015). The Fast Bilateral Solver, *arXiv:1511.03296*. Cited on page 18.

S. Basu, I. Davidson, and K. Wagstaff (2008). Constrained Clustering: Advances in Algorithms, Theory, and Applications. Cited on page 99.

A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei (2015). What's the point: Semantic segmentation with point supervision, *arXiv:1506.02106*. Cited on pages 19 and 64.

M. V. D. Bergh, G. Roig, X. Boix, S. Manen, and L. V. Gool (2013). Online Video SEEDS for Temporal Window Objectness, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on pages 125, 127, 130, and 192.

G. Bertasius, J. Shi, and L. Torresani (2015a). DeepEdge: A Multi-Scale Bifurcated Deep Network for Top-Down Contour Detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 16 and 132.

G. Bertasius, J. Shi, and L. Torresani (2015b). High-for-Low and Low-for-High: Efficient Boundary Detection from Deep Object Features and its Applications to High-Level Vision, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on page 16.

G. Bertasius, J. Shi, and L. Torresani (2016). Semantic Segmentation with Boundary Neural Fields, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 16.

L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr (2016). Fully-Convolutional Siamese Networks for Object Tracking, *arXiv:1606.09549*. Cited on pages 29, 145, 159, and 166.

H. Bilen and A. Vedaldi (2016). Weakly Supervised Deep Detection Networks, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 75.

F. Bookstein (1989). Principal warps: Thin-plate splines and the decomposition of deformations, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 145, 148, and 167.

A. Borji, M.-M. Cheng, H. Jiang, and J. Li (2015). Salient Object Detection: A Benchmark, *IEEE Trans. on Image Processing*. Cited on page 82.

K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan (2017). Unsupervised Pixel-level Domain Adaptation with Generative Adversarial Networks, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on page 195.

L. Breiman (2001). Random Forests, *Machine Learning*. Cited on page 102.

M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool (2009). Robust tracking-by-detection using a detector confidence particle filter, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2009*. Cited on pages 29 and 166.

W. Brendel and S. Todorovic (2009). Video Object Segmentation by Tracking Regions, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2009*. Cited on page 97.

T. Brox and J. Malik (2010). Object segmentation by long term analysis of point trajectories, in *Proc. of the European Conf. on Computer Vision (ECCV) 2010*. Cited on pages 23, 24, 97, 98, 101, 103, 104, 105, 109, 111, 112, 113, 115, 125, 126, 128, and 139.

I. Budvytis, V. Badrinarayanan, and R. Cipolla (2011). Semi-Supervised Video Segmentation Using Tree Structured Graphical Models, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 35, pp. 2751–2764. Cited on page 28.

T. Bühler and M. Hein (2009). Spectral Clustering Based on the Graph p-Laplacian, in *Proc. of the International Conf. on Machine learning (ICML) 2009*. Cited on pages 97, 100, and 122.

S. Caelles, Y. Chen, J. Pont-Tuset, and L. V. Gool (2017a). Semantically-Guided Video Object Segmentation, *arxiv: 1704.01926*. Cited on page 30.

S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. V. Gool (2017b). One-Shot Video Object Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on pages 23, 29, 30, 145, 159, 160, 163, 166, 173, 174, 180, 181, 183, 185, 189, and 193.

H. Caesar, J. R. R. Uijlings, and V. Ferrari (2015). Joint Calibration for Semantic Segmentation, in *Proc. of the British Machine Vision Conf. (BMVC) 2015*. Cited on page 18.

J. Canny (1986). A Computational Approach to Edge Detection, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 15, 37, 39, 40, and 190.

C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, D. Ramanan, and T. Huang (2015). Look and Think Twice: Capturing Top-Down Visual Attention with Feedback Convolutional Neural Networks, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 17 and 75.

J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik (2016). Human Pose Estimation with Iterative Error Feedback, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 68.

J. Carreira, R. Caseiro, J. P. Batista, and C. Sminchisescu (2012). Semantic Segmentation with Second-Order Pooling, in *Proc. of the European Conf. on Computer Vision (ECCV) 2012*. Cited on page 17.

J. Carreira and C. Sminchisescu (2012). CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34(7), pp. 1312–1328. Cited on page 17.

S. Chandra and I. Kokkinos (2016). Fast, Exact and Multi-Scale Inference for Semantic Image Segmentation with Deep Gaussian CRFs. Cited on page 19.

J. Chang, D. Wei, and J. W. Fisher (2013). A Video Representation Using Temporal Superpixels, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on pages 97, 100, 109, 113, 125, 126, 127, 130, and 192.

A. Chaudhry, P. K. Dokania, and P. H. Torr (2017). Discovering Class-Specific Pixels for Weakly-Supervised Semantic Segmentation, *arxiv: 1707.05821*. Cited on pages 19 and 21.

J. Chen, S. Paris, and F. Durand (2007). Real-time Edge-aware Image Processing with the Bilateral Grid, *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH)*, vol. 26(3). Cited on page 28.

L. Chen, G. Papandreou, F. Schroff, and H. Adam (2017a). Rethinking Atrous Convolution for Semantic Image Segmentation, *arxiv: 1706.05587*. Cited on pages 19, 30, 181, and 194.

L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille (2016a). Semantic Image Segmentation with Task-Specific Edge Detection Using CNNs and a Discriminatively Trained Domain Transform, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 16.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille (2015). Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs, in *Proc. of the International Conf. on Learning Representations (ICLR) 2015*. Cited on pages 18, 19, 20, 27, 57, 62, 63, and 64.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille (2016b). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, *arXiv:1606.00915*. Cited on pages 18, 19, 23, 64, 67, 68, 77, 87, 89, 143, 145, 147, 148, 149, 163, 176, 194, and 197.

W. Chen, H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-Or, and B. Chen (2016c). Synthesizing training images for boosting human 3D pose estimation, in *3D Vision (3DV) 2016*. Cited on page 161.

Y. Chen, W. Chen, Y. Chen, B. Tsai, Y. F. Wang, and M. Sun (2017b). No More Discrimination: Cross City Adaptation of Road Scene Segmenters, *arXiv:1704.08509*. Cited on page 8.

H.-T. Cheng and N. Ahuja (2012). Exploiting nonlocal spatiotemporal structure for video segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on pages 97 and 111.

J. Cheng, S. Liu, Y.-H. Tsai, W.-C. Hung, S. Gupta, J. Gu, J. Kautz, S. Wang, and M.-H. Yang (2017). Learning to Segment Instances in Videos with Spatial Propagation Network, *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*. Cited on page 186.

M. Cheng, V. Prisacariu, S. Zheng, P. Torr, and C. Rother (2015a). DenseCut: Densely Connected CRFs for Realtime GrabCut, *Computer Graphics Forum*. Cited on pages 22, 41, and 59.

M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu (2015b). Global Contrast based Salient Region Detection, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 148.

P. Chockalingam, S. N. Pradeep, and S. Birchfield (2009). Adaptive fragments-based tracking of non-rigid objects using level sets, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2009*. Cited on page 29.

J. Corso, E. Sharon, S. Dube, S. El-Saden, U. Sinha, and A. Yuille (2008). Efficient multilevel brain tumor segmentation with integrated bayesian model classification, *Trans. Med. Imaging*. Cited on page 115.

J. M. Coughlan, A. L. Yuille, S. Konishi, and S. C. Zhu (2003). Statistical Edge Detection: Learning and Evaluating Edge Cues, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 25, pp. 57–74. Cited on page 15.

C. Couprie, L. J. Grady, L. Najman, and H. Talbot (2011). Power Watershed: A Unifying Graph-Based Optimization Framework., *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 111.

A. Criminisi, P. Perez, and K. Toyama (2004). Region Filling and Object Removal by Exemplar-based Image Inpainting, *IEEE Trans. on Image Processing*. Cited on page 167.

A. Criminisi, J. Shotton, and E. Konukoglu (2012). Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning, in *Foundations and Trends in Computer Graphics and Vision 2012*. Cited on pages 102 and 106.

J. Dai, K. He, Y. Li, S. Ren, and J. Sun (2016a). Instance-sensitive Fully Convolutional Networks, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on pages 22 and 196.

J. Dai, K. He, and J. Sun (2015a). Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 5, 19, 20, 61, 62, 64, 66, and 67.

J. Dai, K. He, and J. Sun (2015b). Convolutional Feature Masking for Joint Object and Stuff Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 18, 22, 23, and 125.

J. Dai, K. He, and J. Sun (2016b). Instance-Aware Semantic Segmentation via Multi-Task Network Cascades, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 22.

J. Dai, Y. Li, K. He, and J. Sun (2016c). R-FCN: Object Detection via Region-based Fully Convolutional Networks, in *Advances in Neural Information Processing Systems (NIPS) 2016*. Cited on pages 22 and 198.

M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg (2015). Convolutional features for correlation filter based visual tracking, in *ICCV Workshops 2015*. Cited on page 29.

M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg (2016). Beyond correlation filters: Learning continuous convolution operators for visual tracking, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on pages 29, 144, and 146.

M. H. Debidatta Dwibedi, Ishan Misra (2017). Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2017*. Cited on page 195.

M. Defferrard, X. Bresson, and P. Vandergheynst (2016). Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering, in *Advances in Neural Information Processing Systems (NIPS) 2016*. Cited on page 198.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). ImageNet: A Large-Scale Hierarchical Image Database, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2009*. Cited on page 87.

I. S. Dhillon, Y. Guan, and B. Kulis (2007). Weighted Graph Cuts Without Eigenvectors A Multilevel Approach, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 122.

X. Di, H. Chang, and X. Chen (2012). Multi-layer Spectral Clustering for Video Segmentation, in *Proc. of the Asian Conf. on Computer Vision (ACCV) 2012*. Cited on page 115.

C. Doersch, A. Gupta, and A. A. Efros (2015). Unsupervised Visual Representation Learning by Context Prediction, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on page 195.

P. Dollár (). *Piotr's Computer Vision Matlab Toolbox (PMT)*, `http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html`. Cited on page 117.

P. Dollár and C. L. Zitnick (2015). Fast Edge Detection using Structured Forests, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 15, 36, 37, 38, 40, 109, 127, 130, 131, 132, 140, and 190.

A. Dosovitskiy, P. Fischer, 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 *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on page 161.

S. Duffner and C. Garcia (2013). PixelTrack: A Fast Adaptive Algorithm for Tracking Non-rigid Objects, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on page 29.

T. Durand, T. Mordan, N. Thome, and M. Cord (2017). WILDCAT: Weakly Supervised Learning of Deep ConvNets for Image Classification, Pointwise Localization and Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on page 19.

E. Elhamifar and R. Vidal (2009). Sparse subspace clustering, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2009*. Cited on page 24.

L. Ellis and V. Zografos (2013). Online Learning for Fast Segmentation of Moving Objects, in *Proc. of the Asian Conf. on Computer Vision (ACCV) 2013*. Cited on page 196.

A. P. Eriksson, C. Olsson, and F. Kahl (2007). Normalized Cuts Revisited: A Reformulation for Segmentation with Linear Grouping Constraints., in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2007*. Cited on page 99.

M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman (2015). The Pascal Visual Object Classes Challenge: A Retrospective, *International Journal of Computer Vision (IJCV)*. Cited on pages 4, 17, 36, 37, 38, 55, 62, 148, 165, 189, 190, 191, and 198.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman (). *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*, *http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html*. Cited on pages 77, 82, 86, and 87.

A. Faktor and M. Irani (2014). Video Segmentation by Non-Local Consensus voting, in *Proc. of the British Machine Vision Conf. (BMVC) 2014*. Cited on pages 26, 153, 173, and 181.

Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen (2015). JumpCut: Non-Successive Mask Transfer and Interpolation for Video Cutout, *SIGGRAPH Asia*. Cited on page 31.

C. Farabet, C. Couprie, L. Najman, and Y. LeCun (2013). Learning Hierarchical Features for Scene Labeling, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 35(8), pp. 1915–1929. Cited on page 18.

A. Fathi, M.-F. Balcan, X. Ren, and J. M. Rehg (2011). Combining Self Training and Active Learning for Video Segmentation, in *Proc. of the British Machine Vision Conf. (BMVC) 2011*. Cited on page 28.

M. Fayyaz, M. H. Saffar, M. Sabokrou, M. Fathy, and R. Klette (). STFCN: Spatio-Temporal FCN for Semantic Video Segmentation, *arXiv:1608.05971*. Cited on page 9.

P. F. Felzenszwalb and D. P. Huttenlocher. (2004). Efficient Graph-Based Image Segmentation, *International Journal of Computer Vision (IJCV)*. Cited on pages 4, 16, 35, 39, 40, 41, 42, and 190.

C. Fowlkes and J. Malik (2004). How Much Does Globalization Help Segmentation?, Technical report, EECS – UC Berkeley. Cited on pages 25 and 97.

K. Fragkiadaki, P. Arbelaez, P. Felsen, and J. Malik (2015). Learning to segment moving objects in videos, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 25, 26, and 134.

K. Fragkiadaki and J. Shi (2012). Video Segmentation by Tracing Discontinuities in a Trajectory Embedding, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on pages 6, 23, 24, 97, 109, 111, 115, and 125.

B. Fulkerson, A. Vedaldi, and S. Soatto (2009). Class segmentation and object localization with superpixel neighborhoods, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2009*. Cited on page 17.

F. Galasso, R. Cipolla, and B. Schiele (2012). Video Segmentation with superpixels, in *Proc. of the Asian Conf. on Computer Vision (ACCV) 2012*. Cited on pages 6, 15, 23, 24, 25, 97, 98, 99, 100, 101, 102, 104, 105, 106, 107, 111, 112, 113, 115, 116, 117, 118, 121, 122, 123, 136, and 137.

F. Galasso, M. Keuper, T. Brox, and B. Schiele (2014). Spectral Graph Reduction for Efficient Image and Streaming Video Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 6, 23, 25, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 109, 110, 111, 112, 113, 115, 117, 119, 120, 121, 122, 123, 124, 125, 127, 135, 136, 137, 138, 139, 140, and 192.

F. Galasso, N. S. Nagaraja, T. Z. Cardenas, T. Brox, and B. Schiele (2013). A Unified Video Segmentation Benchmark: Annotation, Metrics and Analysis, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on pages 6, 24, 25, 35, 98, 101, 103, 104, 105, 107, 110, 115, 118, 120, 121, 122, 123, 124, 126, 127, 128, 129, 130, 132, 136, 137, 138, 139, 140, and 192.

Y. Ganin and V. Lempitsky (2014). N4-Fields: Neural Network Nearest Neighbor Fields for Image Transforms, in *Proc. of the Asian Conf. on Computer Vision (ACCV) 2014*. Cited on page 16.

Y. Ganin and V. Lempitsky (2015). Unsupervised Domain Adaptation by Backpropagation, in *Proc. of the International Conf. on Machine learning (ICML) 2015*. Cited on page 195.

G. Georgakis, A. Mousavian, A. C. Berg, and J. Kosecka (2017). Synthesizing Training Data for Object Detection in Indoor Scenes, *arXiv:1702.07836*. Cited on pages 161 and 195.

M. George (2015). Image Parsing with a Wide Range of Classes and Scene-Level Context, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 17.

G. Ghiasi and C. C. Fowlkes (2016). Laplacian Reconstruction and Refinement for Semantic Segmentation, *arxiv:1605.02264*. Cited on page 19.

R. Girshick (2015). Fast R-CNN, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 22, 35, 41, 42, 44, 66, 68, and 190.

R. Girshick, J. Donahue, T. Darrell, and J. Malik (2014). Rich feature hierarchies for accurate object detection and semantic segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 18 and 22.

X. Glorot and Y. Bengio (2010). Understanding the difficulty of training deep feedforward neural networks, in *Proc. of the International Conf. on Artificial Intelligence and Statistics (AISTATS) 2010*. Cited on page 172.

M. Godec, P. M. Roth, and H. Bischof (2011). Hough-based Tracking of Non-rigid Objects, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2011*. Cited on page 29.

S. Gould, R. Fulton, and D. Koller (2009). Decomposing a scene into geometric and semantically consistent regions, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2009*. Cited on pages 17 and 196.

P. Goyal, Z. Hu, X. Liang, C. Wang, and E. P. Xing (2017). Nonparametric Variational Auto-encoders for Hierarchical Representation Learning, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2017*. Cited on page 195.

M. Grundmann, V. Kwatra, M. Han, and I. Essa (2010). Efficient Hierarchical Graph-Based Video Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2010*. Cited on pages 23, 24, 97, 100, 104, 107, 109, 111, 112, 113, 121, 122, 123, 125, 126, 127, and 137.

A. Gupta, A. Vedaldi, and A. Zisserman (2016). Synthetic data for text localisation in natural images, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 161.

S. Hallman and C. Fowlkes (2015). Oriented Edge Forests for Boundary Detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 15, 38, and 119.

B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik (2011). Semantic Contours from Inverse Detectors, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2011*. Cited on pages 4, 16, 17, 36, 37, 38, 51, 52, 53, 62, 68, 87, and 190.

B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik (2014). Simultaneous Detection and Segmentation, in *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on pages 18, 22, and 68.

B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik (2015). Hypercolumns for object segmentation and fine-grained localization, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 22, 67, 68, and 146.

Z. Hayder, X. He, and M. Salzmann (2017). Shape-aware Instance Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on page 16.

K. He, G. Gkioxari, P. Dollár, and R. B. Girshick (2017). Mask R-CNN, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2017*. Cited on pages 22, 189, 196, and 198.

K. He, X. Zhang, S. Ren, and J. Sun (2016). Deep Residual Learning for Image Recognition, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on pages 18, 21, and 30.

X. He, R. S. Zemel, and M. A. Carreira-Perpinan (2004). Multiscale conditional random fields for image labeling, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2004*. Cited on page 17.

M. Hein and T. Bühler (2010). An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse pca, in *Advances in Neural Information Processing Systems (NIPS) 2010*. Cited on pages 24, 97, 100, 101, and 122.

M. Hein and S. Setzer (2011). Beyond Spectral Clustering - Tight Relaxations of Balanced Graph Cuts, in *Advances in Neural Information Processing Systems (NIPS) 2011*. Cited on pages 99, 101, 103, 104, and 106.

D. Held, S. Thrun, and S. Savarese (2016). Learning to Track at 100 FPS with Deep Regression Networks, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on pages 29, 145, 159, and 166.

J. F. Henriques, R. Caseiro, P. Martins, and J. Batista (2012). Exploiting the Circulant Structure of Tracking-by-detection with Kernels, in *Proc. of the European Conf. on Computer Vision (ECCV) 2012*. Cited on pages 29, 162, and 166.

J. Hoffman, D. Wang, F. Yu, and T. Darrell (2016). Fcns in the wild: Pixel-level adversarial and constraint-based adaptation, *arXiv:1612.02649*. Cited on page 8.

D. Hoiem, A. A. Efros, and M. Hebert (2007). Recovering Surface Layout from an Image, *International Journal of Computer Vision (IJCV)*. Cited on page 113.

S. Hong, H. Noh, and B. Han (2015). Decoupled deep neural network for semi-supervised semantic segmentation, in *Advances in Neural Information Processing Systems (NIPS) 2015*. Cited on pages 18 and 19.

S. Hong, J. Oh, H. Lee, and B. Han (2016). Learning Transferrable Knowledge for Semantic Segmentation with Deep Convolutional Neural Network. Cited on page 19.

S. Hong, D. Yeo, S. Kwak, H. Lee, and B. Han (2017). Weakly Supervised Semantic Segmentation using Web-Crawled Videos, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on pages 19, 20, and 195.

J. Hosang, R. Benenson, P. Dollár, and B. Schiele (2015). What makes for effective detection proposals?, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 22 and 35.

Q. Hou, P. K. Dokania, D. Massiceti, Y. Wei, M. Cheng, and P. H. S. Torr (2016). Mining Pixels: Weakly Supervised Semantic Segmentation Using Image Labels, *arxiv: 1612.02101*. Cited on page 20.

X. Hou, A. Yuille, and C. Koch (2013). Boundary Detection Benchmarking: Beyond F-Measures, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on page 39.

P.-J. Hsieh, E. Vul, and N. Kanwisher (2010). Recognition alters the spatial pattern of FMRI activation in early retinotopic cortex., *Journal of neurophysiology*, vol. 103 3, pp. 1501–7. Cited on page 15.

A. Humayun, F. Li, and J. M. Rehg (2014). RIGOR: Recycling Inference in Graph Cuts for generating Object Regions, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 125 and 132.

A. Humayun, F. Li, and J. M. Rehg (2015). The Middle Child Problem Revisiting Parametric Min-Cut and Seeds for Object Proposals, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 125 and 132.

E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox (2017). FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on pages 163, 172, 175, and 196.

P. Isola, D. Zoran, D. Krishnan, , and E. H. Adelson (2014). Crisp Boundary Detection Using Pointwise Mutual Information, in *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on pages 16, 40, 109, 127, and 130.

A. Jain, S. Chatterjee, and R. Vidal (2013). Coarse-to-fine Semantic Video Segmentation using Supervoxel Trees, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on pages 23, 109, 111, 125, and 127.

M. Jain, J. C. van Gemert, H. Jégou, P. Bouthemy, and C. Snoek (2014). Action Localization with Tubelets from Motion, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 25 and 26.

S. D. Jain and K. Grauman (2014). Supervoxel-consistent foreground propagation in video, in *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on pages 8, 27, 28, 150, and 170.

S. D. Jain and K. Grauman (2016). Click Carving: Segmenting Objects in Video with Point Clicks., in *Conf. on Human Computation and Crowdsourcing 2016*. Cited on pages 23 and 31.

S. D. Jain, B. Xiong, and K. Grauman (2017). FusionSeg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos, *arXiv:1701.05384*. Cited on pages 23, 27, 30, 163, 173, 181, 188, and 189.

V. Jain, S. C. Turaga, K. L. Briggman, M. Helmstaedter, W. Denk, and H. S. Seung (2011). Learning to Agglomerate Superpixel Hierarchies., in *Advances in Neural Information Processing Systems (NIPS) 2011*. Cited on page 99.

V. Jampani, R. Gadde, and P. V. Gehler (2016a). Video Propagation Networks, *arXiv:1612.05478*. Cited on pages 30, 159, 163, 173, and 181.

V. Jampani, M. Kiefel, and P. V. Gehler (2016b). Learning Sparse High Dimensional Filters: Image Filtering, Dense CRFs and Bilateral Neural Networks, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 19.

T. Jebara and S. Chang (2009). Graph construction and bmatching for semi-supervised learning, in *Proc. of the International Conf. on Machine learning (ICML) 2009*. Cited on page 111.

T. Jebara and V. Shchogolev (2006). B-Matching for Spectral Clustering, in *Proc. of the European Conf. on Machine learning (ECML) 2006*. Cited on page 111.

H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li (2013). Salient object detection: A discriminative regional feature integration approach, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on page 82.

B. Jin, M. V. Ortiz Segovia, and S. Susstrunk (2017). Webly Supervised Semantic Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on pages 19, 20, and 195.

N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu (2016). Video Pixel Networks, *arXiv:1610.00527*. Cited on page 9.

S. D. Kamvar, D. Klein, and C. D. Manning (2003). Spectral learning, in *Proc. of the International Joint Conf. on Artificial Intelligence (IJCAI) 2003*. Cited on page 99.

A. Kannan, N. Jojic, and B. J. Frey (2005). Generative Model for Layers of Appearance and Deformation, in *Proc. of the International Conf. on Artificial Intelligence and Statistics (AISTATS) 2005*. Cited on page 97.

V. Kantorov, M. Oquab, M. Cho, and I. Laptev (2016). ContextLocNet: Context-Aware Deep Network Models for Weakly Supervised Localization, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on page 75.

M. Keuper (2017). Higher-Order Minimum Cost Lifted Multicuts for Motion Segmentation, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2017*. Cited on page 24.

M. Keuper, B. Andres, and T. Brox (2015). Motion Trajectory Segmentation via Minimum Cost Multicuts, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on page 24.

M. Keuper and T. Brox (2016). Point-wise mutual information-based video segmentation with high temporal consistency, *arXiv:1606.02467*. Cited on pages 6, 24, 125, and 137.

A. Khoreva, R. Benenson, F. Galasso, M. Hein, and B. Schiele (2016a). Improved Image Boundaries for Better Video Segmentation, in *ECCV Workshops 2016*. Cited on page 12.

A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele (2017a). Simple Does It: Weakly Supervised Instance and Semantic Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on pages 12 and 87.

A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele (2017b). Lucid Data Dreaming for Object Tracking, *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*. Cited on page 13.

A. Khoreva, R. Benenson, M. Omran, M. Hein, and B. Schiele (2016b). Weakly Supervised Object Boundaries, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on pages 11 and 67.

A. Khoreva, F. Galasso, M. Hein, and B. Schiele (2014). Learning Must-Link Constraints for Video Segmentation based on Spectral Clustering, in *Proc. of the German Conf. on Pattern Recognition (GCPR) 2014*. Cited on pages 12, 121, 122, and 128.

A. Khoreva, F. Galasso, M. Hein, and B. Schiele (2015). Classifier Based Graph Construction for Video Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 12, 23, 24, 25, 125, 137, and 138.

S. Kim, S. Nowozin, P. Kohli, and C. D. Yoo (2013). Task-Specific Image Partitioning, *IEEE Trans. on Image Processing*. Cited on pages 112 and 121.

T. N. Kipf and M. Welling (2017). Semi-Supervised Classification with Graph Convolutional Networks, in *Proc. of the International Conf. on Learning Representations (ICLR) 2017*. Cited on page 198.

A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother (2017). InstanceCut: from Edges to Instances with Multicut, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on pages 15 and 16.

J. Kittler (1983). On the accuracy of the Sobel edge detector, *Image and Vision Computing*, vol. 1, pp. 37 – 42. Cited on page 15.

I. Kokkinos (2016). Pushing the Boundaries of Boundary Detection using Deep Learning, in *Proc. of the International Conf. on Learning Representations (ICLR) 2016*. Cited on pages 15, 16, 19, and 62.

A. Kolesnikov and C. Lampert (2016a). Improving Weakly-Supervised Object Localization by Micro-Annotation, in *Proc. of the British Machine Vision Conf. (BMVC) 2016*. Cited on pages 20, 74, and 92.

A. Kolesnikov and C. H. Lampert (2016b). Seed, Expand and Constrain: Three Principles for Weakly-Supervised Image Segmentation, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on pages 19, 20, 77, 81, 87, 92, and 189.

V. Kolmogorov and R. Zabih (2004). What Energy Functions Can Be Minimized via Graph Cuts?., *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 18.

Z. Kourtzi and N. Kanwisher (2001). Representation of perceived object shape by the human lateral occipital complex., *Science*, vol. 293 5534, pp. 1506–9. Cited on page 15.

P. Krähenbühl and V. Koltun (2011). Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials, in *Advances in Neural Information Processing Systems (NIPS) 2011*. Cited on pages 3, 17, 18, 20, 27, 57, 62, 80, 81, 89, 151, 165, 172, and 176.

P. Krähenbühl and V. Koltun (2014). Geodesic Object Proposals, in *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on pages 22, 125, and 132.

P. Krähenbühl and V. Koltun (2015). Learning to propose objects, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 22.

M. Kristan, J. Matas, *et al.* (2015). The Visual Object Tracking VOT2015 challenge results, in *ICCV Workshops 2015*. Cited on page 159.

M. Kristan, J. Matas, *et al.* (2016). The Visual Object Tracking VOT2016 Challenge Results, in *ECCV Workshops 2016*. Cited on page 159.

M. Kristan, R. Pflugfelder, *et al.* (2014). The Visual Object Tracking VOT2014 challenge results, in *ECCV Workshops 2014*. Cited on pages 29, 159, and 166.

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 17 and 143.

M. P. Kumar, P. Torr, and A. Zisserman (2008). Learning Layered Motion Segmentations of Video, in *International Journal of Computer Vision (IJCV) 2008*. Cited on page 97.

A. Kundu, V. Vineet, and V. Koltun (2016). Feature Space Optimization for Semantic Video Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 198.

L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr (2009). Associative hierarchical CRFs for object class image segmentation, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2009*. Cited on page 17.

J. D. Lafferty, A. McCallum, and F. C. N. Pereira (2001). Conditional random fields: probabilistic models for segmenting and labeling sequence data, in *Proc. of the International Conf. on Machine learning (ICML) 2001*. Cited on page 81.

T.-N. Le, K.-T. Nguyen, M.-H. Nguyen-Phan, T.-V. Ton, T.-A. N. (2), X.-S. Trinh, Q.-H. Dinh, V.-T. Nguyen, A.-D. Duong, A. Sugimoto, T. V. Nguyen, and M.-T. Tran (2017). Instance Re-Identification Flow for Video Object Segmentation, *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*. Cited on page 186.

Y. J. Lee, J. Kim, and K. Grauman (2011). Key-Segments for Video Object Segmentation, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2011*. Cited on pages 24 and 26.

V. Lempitsky, P. Kohli, C. Rother, and T. Sharp (2009). Image segmentation with a bounding box prior, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2009*. Cited on page 22.

T. Leung and J. Malik (2001). Representing and recognizing the visual appearance of materials using three-dimensional textons, *International Journal of Computer Vision (IJCV)*. Cited on page 113.

A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi (2009). TurboPixels: Fast Superpixels Using Geometric Flows, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 127.

J. Lezama, K. Alahari, J. Sivic, and I. Laptev (2011). Track to the future: Spatio-temporal video segmentation with long-range motion cues, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2011*. Cited on pages 24, 97, and 112.

F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg (2013). Video Segmentation by Tracking Many Figure-Ground Segments, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on pages 8, 26, 97, 109, 150, 159, and 170.

G. Li and Y. Yu (2016). Deep Contrast Learning for Salient Object Detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 87.

K. Li, B. Hariharan, and J. Malik (2016a). Iterative Instance Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 22.

R. Li, M. Tapaswi, R. Liao, J. Jia, R. Urtasun, and S. Fidler (2017a). Situation Recognition with Graph Neural Networks, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2017*. Cited on page 198.

X. Li, Y. Qi, Z. Wang, K. Chen, Z. Liu, J. Shi, P. Luo, C. C. Loy, and X. Tang (2017b). Video Object Segmentation with Re-identification, *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*. Cited on pages 30, 185, 186, 187, and 197.

X. Li, L. Zhao, L. Wei, M. H. Yang, F. Wu, Y. Zhuang, H. Ling, and J. Wang (2016b). DeepSaliency: Multi-Task Deep Neural Network Model for Salient Object Detection, *IEEE Trans. on Image Processing*. Cited on page 87.

Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille (2014). The Secrets of Salient Object Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2014*. Cited on pages 82 and 148.

Y. Li, M. Paluri, J. M. Rehg, and P. Dollár (2015). Unsupervised Learning of Edges, in *arXiv:1511.04166 2015*. Cited on pages 16 and 189.

Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei (2017c). Fully Convolutional Instance-aware Semantic Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on page 22.

Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel (2016c). Gated Graph Sequence Neural Networks, in *Proc. of the International Conf. on Learning Representations (ICLR) 2016*. Cited on page 198.

Z. Li, J. Liu, and X. Tang (2009). Constrained clustering via spectral regularization, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2009*. Cited on page 99.

X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan (2016). Semantic Object Parsing with Graph LSTM, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on page 198.

J. J. Lim, C. L. Zitnick, and P. Dollár (2013). Sketch Tokens: A Learned Mid-level Representation for Contour and Object Detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on page 15.

D. Lin, J. Dai, J. Jia, K. He, and J. Sun (2016a). ScribbleSup: Scribble-Supervised Convolutional Networks for Semantic Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 19.

G. Lin, A. Milan, C. Shen, and I. D. Reid (2016b). RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation, *arXiv:1611.06612*. Cited on page 19.

G. Lin, C. Shen, A. van dan Hengel, and I. Reid (2016c). Efficient Piecewise Training of Deep Structured Models for Semantic Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 19.

T. 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 *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on pages 4, 8, 17, 37, 38, 55, 62, 66, 82, 148, 189, 191, and 198.

B. Liu and X. He (2015). Multiclass Semantic Video Segmentation with Object-level Active Inference, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 198.

T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum (2011). Learning to detect a salient object, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 33(2), pp. 353–367. Cited on page 87.

W. Liu, A. Rabinovich, and A. C. Berg (2015). ParseNet: Looking Wider to See Better, *arxiv:1506.04579*. Cited on pages 18, 19, and 194.

Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang (2017). Deep Learning Markov Random Field for Semantic Segmentation, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on page 19.

J. Long, E. Shelhamer, and T. Darrell (2015). Fully Convolutional Networks for Semantic Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 18, 27, and 189.

M. Long, H. Zhu, J. Wang, and M. I. Jordan (2016). Unsupervised Domain Adaptation with Residual Transfer Networks, in *Advances in Neural Information Processing Systems (NIPS) 2016*. Cited on page 195.

C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang (2015). Hierarchical Convolutional Features for Visual Tracking, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on page 29.

T. Ma and L. J. Latecki (2012). Maximum weight cliques with mutex constraints for video object segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on page 26.

N. Maerki, F. Perazzi, O. Wang, and A. Sorkine-Hornung (2016). Bilateral Space Video Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on pages 23, 27, 28, 143, 150, 153, 166, 173, 181, and 183.

M. Maier, U. von Luxburg, and M. Hein (2009). Influence of graph construction on graph-based clustering measures, in *Advances in Neural Information Processing Systems (NIPS) 2009*. Cited on page 111.

M. Maire and S. X. Yu (2013). Progressive Multigrid Eigensolvers for Multiscale Spectral Segmentation, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on pages 6, 97, 98, 101, 109, 111, and 115.

S. Maji, N. K. Vishnoi, and J. Malik (2011). Biased normalized cuts., in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2011*. Cited on page 99.

F. Manessi, A. Rozza, and M. Manzo (2017). Dynamic Graph Convolutional Networks, *arXiv: 1704.06199*. Cited on page 198.

K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. V. Gool (2017). Convolutional Oriented Boundaries: From Image Segmentation to High-Level Tasks, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 16 and 147.

D. Marr and E. Hildreth (1980). Theory of Edge Detection, *Proceedings of the Royal Society of London Series B*, vol. 207, pp. 187–217. Cited on page 15.

D. Martin, C. Fowlkes, D. Tal, and J. Malik (2001). A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2001*. Cited on page 37.

D. R. Martin, C. C. Fowlkes, and J. Malik (2004). Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 26(5), pp. 530–549. Cited on page 15.

N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox (2016). A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on pages 161, 163, and 189.

M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich (2015). Feedforward semantic segmentation with zoom-out features, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 18.

V. Movahedi and J. H. Elder (2010). Design and perceptual validation of performance measures for salient object segmentation, in *CVPR Workshops 2010*. Cited on page 148.

N. Nagaraja, F. Schmidt, and T. Brox (2015). Video Segmentation with Just a Few Strokes, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 23, 27, 28, 31, and 173.

H. Nam, M. Baek, and B. Han (2016). Modeling and propagating cnns in a tree structure for visual tracking, *arXiv:1608.07242*. Cited on page 29.

H. Nam and B. Han (2016). Learning Multi-Domain Convolutional Neural Networks for Visual Tracking, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on pages 29, 144, 145, 146, 159, and 196.

P. Neubert and P. Protzel (2013). Evaluating Superpixels in Video Metrics Beyond Figure-Ground Segmentation, in *Proc. of the British Machine Vision Conf. (BMVC) 2013*. Cited on page 130.

A. Y. Ng, M. Jordan, and Y. Weiss (2001). On spectral clustering: Analysis and an algorithm, in *Advances in Neural Information Processing Systems (NIPS) 2001*. Cited on pages 24, 97, 99, 100, 101, 103, 111, 113, and 115.

D. Nilsson and C. Sminchisescu (2016). Semantic Video Segmentation by Gated Recurrent Flow Propagation, *arXiv: 1612.08871*. Cited on page 198.

H. Noh, S. Hong, and B. Han (2015). Learning deconvolution network for semantic segmentation, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on page 18.

P. Ochs and T. Brox (2012). Higher order motion models and spectral clustering, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on page 24.

P. Ochs, J. Malik, and T. Brox (2014). Segmentation of moving objects by long term video analysis, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 6, 8, 23, 24, 97, 125, and 127.

S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, and B. Schiele (2017). Exploiting Saliency for Object Segmentation from Image Level Labels, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on page 12.

D. Oneata, J. Revaud, J. Verbeek, and C. Schmid (2014). Spatio-Temporal Object Detection Proposals, in *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on pages 26 and 117.

M. Oquab, L. Bottou, L. I, and S. J (2015). Is object localization for free? – Weakly-supervised learning with convolutional neural networks, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 17.

G. Palou and P. Salembier (2013). Hierarchical Video Representation with Trajectory Binary Partition Tree, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on pages 23, 97, 109, 111, 112, 113, 125, and 126.

G. Papandreou, L. Chen, K. Murphy, , and A. L. Yuille (2015). Weakly- and Semi-Supervised Learning of a DCNN for Semantic Image Segmentation, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 5, 19, 20, 62, 64, 66, 67, 92, and 189.

A. Papazoglou and V. Ferrari (2013). Fast object segmentation in unconstrained video, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on pages 26 and 127.

S. Paris (2008). Edge-preserving Smoothing and Mean-shift Segmentation of Video Streams, in *Proc. of the European Conf. on Computer Vision (ECCV) 2008*. Cited on pages 100 and 113.

D. Park and D. Ramanan (2015). Articulated pose estimation with tiny synthetic videos, in *CVPR Workshops 2015*. Cited on pages 161 and 195.

D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan (2016). Learning Features by Watching Objects Move, *arXiv:1612.06370*. Cited on page 166.

D. Pathak, P. Kraehenbuehl, and T. Darrell (2015a). Constrained Convolutional Neural Networks for Weakly Supervised Segmentation, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 19, 20, 74, and 92.

D. Pathak, E. Shelhamer, J. Long, and T. Darrell (2015b). Fully Convolutional Multi-Class Multiple Instance Learning, in *ICLR Workshops 2015*. Cited on pages 19, 20, and 92.

F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung (2017). Learning Video Object Segmentation from Static Images, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on pages 13, 30, 159, 160, 163, 166, 173, 175, 180, 181, 183, and 188.

F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, and A. Sorkine-Hornung (2016). A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on pages 150, 153, 159, 170, and 180.

F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung (2015). Fully Connected Object Proposals for Video Segmentation, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 27, 28, 153, 173, and 181.

P. O. Pinheiro and R. Collobert (2014). Recurrent Convolutional Neural Networks for Scene Labeling, in *Proc. of the International Conf. on Machine learning (ICML) 2014*. Cited on page 18.

P. O. Pinheiro and R. Collobert (2015). From Image-level to Pixel-level Labeling with Convolutional Network, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 17, 19, 20, 74, 75, and 92.

P. O. Pinheiro, R. Collobert, and P. Dollar (2015). Learning to segment object candidates, in *Advances in Neural Information Processing Systems (NIPS) 2015*. Cited on pages 22, 67, 68, and 146.

P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár (2016). Learning to refine object segments, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on pages 22, 67, and 146.

L. Pishchulin, A. Jain, M. Andriluka, T. Thormählen, and B. Schiele (2012). Articulated people detection and pose estimation: Reshaping the future, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on page 161.

T. Pohlen, A. Hermans, M. Mathias, and B. Leibe (2017). Full-Resolution Residual Networks for Semantic Segmentation in Street Scenes, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on pages 19, 181, and 198.

J. Pont-Tuset, P. Arbeláez, J. Barron, F. Marques, and J. Malik (2016). Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 16, 20, 22, 35, 42, 61, 92, 125, 127, 130, 132, 133, 190, and 191.

J. Pont-Tuset and L. V. Gool (2015). Boosting Object Proposals: From Pascal to COCO, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 4, 21, 57, 58, 61, and 71.

J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool (). *DAVIS Challenge on Video Object Segmentation 2017*, `http://davischallenge.org/challenge2017`. Cited on page 185.

J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool (2017). The 2017 DAVIS Challenge on Video Object Segmentation, *arXiv:1704.00675*. Cited on pages 8, 183, 184, 189, 194, and 198.

V. Premachandran, B. Bonev, and A. L. Yuille (2017). PASCAL Boundaries: A Class-Agnostic Semantic Boundary Dataset, in *Proc. IEEE Winter Conf. on Applications of Computer Vision (WACV) 2017*. Cited on page 17.

A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari (2012). Learning object class detectors from weakly annotated video, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on pages 17, 150, 159, and 170.

X. Qi, Z. Liu, J. Shi, H. Zhao, and J. Jia (2016). Augmented Feedback in Semantic Segmentation Under Image Level Supervision, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on pages 20 and 92.

V. Ramanishka, A. Das, J. Zhang, and K. Saenko (2017). Top-Down Visual Saliency Guided by Captions, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on page 197.

S. Rangapuram and M. Hein (2012). Constrained 1-Spectral Clustering, in *Proc. of the International Conf. on Artificial Intelligence and Statistics (AISTATS) 2012*. Cited on pages 98, 99, 100, 101, and 103.

S. H. Raza, M. Grundmann, and I. Essa (2013). Geoemetric Context from Video, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on pages 109 and 117.

S. Ren, K. He, R. Girshick, and J. Sun (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, in *Advances in Neural Information Processing Systems (NIPS) 2015*. Cited on pages 22, 35, 41, and 190.

X. Ren and L. Bo (2012). Discriminatively Trained Sparse Code Gradients for Contour Detection, in *Advances in Neural Information Processing Systems (NIPS) 2012*. Cited on page 109.

X. Ren and J. Malik (2003). Learning a Classification Model for Segmentation, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2003*. Cited on pages 23, 112, 121, 125, and 127.

X. Ren and J. Malik (2007). Tracking as Repeated Figure/Ground Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2007*. Cited on page 29.

M. Reso, J. Jachalsky, B. Rosenhahn, and J. Ostermann (2013). Temporally Consistent Superpixels, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on page 109.

J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid (2015). EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 134, 147, and 175.

S. R. Richter, Z. Hayder, and V. Koltun (2017). Playing for Benchmarks, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2017*. Cited on page 198.

S. R. Richter, V. Vineet, S. Roth, and V. Koltun (2016). Playing for Data: Ground Truth from Computer Games, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on page 161.

A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele (2016). Grounding of Textual Phrases in Images by Reconstruction, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on page 197.

O. Ronneberger, P.Fischer, and T. Brox (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation, in *Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2015*. Cited on pages 18 and 27.

C. Rother, V. Kolmogorov, and A. Blake (2004). Grabcut: Interactive foreground extraction using iterated graph cuts, in *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH) 2004*. Cited on pages 4, 16, 22, 41, 58, 59, 163, 172, 190, and 191.

A. Roy and S. Todorovic (2017). Combining Bottom-Up, Top-Down, and Smoothness Cues for Weakly Supervised Image Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on page 21.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei (2015). ImageNet Large Scale Visual Recognition Challenge, *International Journal of Computer Vision (IJCV)*. Cited on pages 55, 143, 189, and 198.

F. Saleh, M. S. A. Akbarian, M. Salzmann, L. Petersson, S. Gould, and J. M. Alvarez (2016). Built-in Foreground/Background Prior for Weakly-Supervised Semantic Segmentation, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on pages 20, 74, and 92.

A. G. Schwing and R. Urtasun (2015). Fully Connected Deep Structured Networks, *arXiv:1503.02351*. Cited on page 19.

P. Sermanet, C. Lynch, J. Hsu, and S. Levine (2017). Time-Contrastive Networks: Self-Supervised Learning from Multi-View Observation, *arXiv:1704.06888*. Cited on page 189.

A. Shaban, A. Firl, A. Humayun, J. Yuan, X. Wang, P. Lei, N. Dhanda, B. Boots, J. M. Rehg, and F. Li (2017). Multiple-Instance Video Segmentation with Sequence-Specific Object Proposals, *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*. Cited on page 186.

G. Sharir, E. Smolyansky, and I. Friedman (2017). Video Object Segmentation using Tracked Object Proposals, *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*. Cited on page 186.

A. Sharma, O. Tuzel, and D. W. Jacobs (2015). Deep Hierarchical Parsing for Semantic Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 18.

I. Shcherbatyi and B. Andres (2016). Convexification of Learning from Constraints, in *Proc. of the German Conf. on Pattern Recognition (GCPR) 2016*. Cited on page 73.

W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang (2015). DeepContour: A deep convolutional feature learned by positive-sharing loss for contour detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 16.

J. Shi and J. Malik (2000). Normalized Cuts and Image Segmentation, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 24, 97, 99, 100, 101, 103, 109, 111, 113, and 115.

J. Shi, Q. Yan, L. Xu, and J. Jia (2016). Hierarchical Image Saliency Detection on Extended CSSD, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*. Cited on pages 82 and 148.

W. Shimoda and K. Yanai (2016). Distinct class-specific saliency maps for weakly supervised semantic segmentation, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on pages 20, 75, and 92.

J. Shotton, M. Johnson, and R. Cipolla (2008). Semantic texton forests for image categorization and segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2008*. Cited on page 17.

J. Shotton, J. Winn, C. Rother, and A. Criminisi (2009). Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context, *International Journal of Computer Vision (IJCV)*. Cited on page 17.

M. Siam, S. Valipour, M. Jägersand, and N. Ray (). Convolutional Gated Recurrent Networks for Video Segmentation, *arXiv:1611.05435*. Cited on pages 9 and 197.

K. Simonyan, A. Vedaldi, and A. Zisserman (2014). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, in *ICLR Workshops 2014*. Cited on pages 41, 75, 76, and 80.

K. Simonyan and A. Zisserman (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition, in *Proc. of the International Conf. on Learning Representations (ICLR) 2015*. Cited on pages 17, 18, 39, 44, 62, 77, 80, 87, 148, 172, and 184.

L. Smith and M. Gasser (2005). The Development of Embodied Cognition: Six Lessons from Babies, *Artificial Life*, vol. 11(1-2), pp. 13–29. Cited on page 199.

R. Socher, C. C.-Y. Lin, A. Y. Ng, and C. D. Manning (2011). Parsing Natural Scenes and Natural Language with Recursive Neural Networks, in *Proc. of the International Conf. on Machine learning (ICML) 2011*. Cited on page 196.

N. Souly, C. Spampinato, and M. Shah (2017). Semi and Weakly Supervised Semantic Segmentation Using Generative Adversarial Network, *arxiv: 1703.09695*. Cited on pages 19 and 21.

T. V. Spina and A. X. Falcão (2016). FOMTrace: Interactive Video Segmentation By Image Graphs and Fuzzy Object Models, *arXiv:1606.03369*. Cited on pages 23 and 31.

J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller (2015). Striving for Simplicity: The All Convolutional Net, in *ICLR Workshops 2015*. Cited on pages 20, 75, 76, and 80.

J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum (2004). Poisson Matting, in *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH) 2004*. Cited on page 167.

N. Sundaram, T. Brox, and K. Keutzer (2010). Dense point trajectories by GPU-accelerated large displacement optical flow, in *Proc. of the European Conf. on Computer Vision (ECCV) 2010*. Cited on pages 24 and 114.

N. Sundaram and K. Keutzer (2011). Long term video segmentation through pixel level spectral clustering on GPUs, in *ICCV Workshops 2011*. Cited on pages 25, 97, 98, 101, 109, 111, and 115.

P. Sundberg, T.Brox, M. Maire, P. Arbelaez, and J. Malik (2011). Occlusion boundary detection and figure/ground assignment from optical flow, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2011*. Cited on pages 97, 101, 105, 115, and 128.

K. Tang, V. Ramanathan, L. Fei-fei, and D. Koller (2012). Shifting Weights: Adapting Object Detectors from Image to Video, in *Advances in Neural Information Processing Systems (NIPS) 2012*. Cited on page 166.

M. Tang, I. Ben Ayed, D. Marin, and Y. Boykov (2015). Secrets of GrabCut and Kernel K-means., in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 22 and 59.

M. Tang, D. Marin, I. Ben Ayed, and Y. Boykov (2016). Normalized Cut Meets MRF, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on page 163.

S. Tang, M. Andriluka, A. Milan, K. Schindler, S. Roth, and B. Schiele (2013). Learning people detectors for tracking in crowded scenes, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on page 161.

T. Taniai, Y. Matsushita, and T. Naemura (2015). Superdifferential Cuts for Binary Energies, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 22.

R. Tao, E. Gavves, and A. W. Smeulders (2016). Siamese Instance Search for Tracking, *arXiv:1605.05863*. Cited on page 29.

E. Taralova, F. D. la Torre, and M. Hebert (2014). Motion Words for Videos, in *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on pages 23, 109, and 125.

B. Taylor, V. Karasev, and S. Soatto (2015). Causal video object segmentation from persistence of occlusions, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 26.

C. Taylor (2013). Towards Fast and Accurate Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on page 97.

E. Teh, M. Rochan, and Y. Wang (2016). Attention Networks for Weakly Supervised Object Localization, in *Proc. of the British Machine Vision Conf. (BMVC) 2016*. Cited on page 75.

P. Tokmakov, K. Alahari, and C. Schmid (2016). Weakly-Supervised Semantic Segmentation using Motion Cues, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on pages 19 and 20.

P. Tokmakov, K. Alahari, and C. Schmid (2017a). Learning Motion Patterns in Videos, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on pages 27, 173, 181, and 189.

P. Tokmakov, K. Alahari, and C. Schmid (2017b). Learning Video Object Segmentation with Visual Memory, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2017*. Cited on pages 9, 23, 27, 189, and 197.

D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri (2015). Learning Spatiotemporal Features with 3D Convolutional Networks, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 9 and 197.

D. Tsai, M. Flagg, and J. M. Rehg (2010). Motion Coherent Tracking with Multi-label MRF optimization, in *British Machine Vision Conference, BMVC 2010, Aberystwyth, UK, August 31 - September 3, 2010. Proceedings 2010*. Cited on pages 27 and 28.

Y.-H. Tsai, M.-H. Yang, and M. J. Black (2016). Video Segmentation via Object Flow, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on pages 23, 27, 28, 143, 149, 150, 153, 173, 181, and 183.

Z. Tu and X. Bai (2010). Auto-Context and Its Application to High-Level Vision Tasks and 3D Brain Image Segmentation, *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32(10), pp. 1744–1757. Cited on page 17.

S. C. Turaga, K. L. Briggman, M. Helmstaedter, W. Denk, and H. S. Seung (2009). Maximin affinity learning of image segmentation, in *Advances in Neural Information Processing Systems (NIPS) 2009*. Cited on pages 23, 112, 121, 125, and 127.

E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell (2017). Adversarial Discriminative Domain Adaptation, *arXiv: 1702.05464*. Cited on page 195.

J. Uijlings and V. Ferrari (2015). Situational Object Boundary Detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 17, 37, 44, 45, 52, and 53.

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 22, 26, 35, 42, and 190.

S. van Dongen (2008). Graph Clustering Via a Discrete Uncoupling Process., *SIAM J. Matrix Analysis Applications*. Cited on page 122.

G. Varol, I. Laptev, and C. Schmid (2016). Long-term Temporal Convolutions for Action Recognition, *arXiv:1604.04494*. Cited on pages 9 and 197.

G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid (). Learning from Synthetic Humans, *arXiv:1701.01370*. Cited on page 161.

A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller (2010). Multiple Hypothesis Video Segmentation from Superpixel Flows, in *Proc. of the European Conf. on Computer Vision (ECCV) 2010*. Cited on pages 97, 100, 113, and 127.

R. Vemulapalli, O. Tuzel, M. Y. Liu, and R. Chellappa (2016). Gaussian Conditional Random Field Network for Semantic Segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 19.

A. Vezhnevets, V. Ferrari, and J. Buhmann (2011). Weakly Supervised Semantic Segmentation with a Multi-image Model, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2011*. Cited on pages 17 and 21.

P. Voigtlaender and B. Leibe (2017a). Online Adaptation of Convolutional Neural Networks for the 2017 DAVIS Challenge on Video Object Segmentation, *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*. Cited on page 186.

P. Voigtlaender and B. Leibe (2017b). Online Adaptation of Convolutional Neural Networks for Video Object Segmentation, *arxiv: 1706.09364*. Cited on pages 30, 188, 193, and 196.

T. Vojir and J. Matas (2017). Pixel-Wise Object Segmentations for the VOT 2016 Dataset, Research report. Cited on page 159.

U. von Luxburg (2007). A tutorial on spectral clustering, *Statistics and Computing*, vol. 17(4), pp. 395–416. Cited on pages 24, 97, 101, and 115.

K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl (2001). Constrained K-means Clustering with Background Knowledge, in *Proc. of the International Conf. on Machine learning (ICML) 2001*. Cited on page 99.

C. Wang, W. Ren, K. Huang, and T. Tan (2014a). Weakly supervised object localization with latent category learning, in *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on page 17.

H. Wang, T. Raiko, L. Lensu, T. Wang, and J. Karhunen (2016). Semi-Supervised Domain Adaptation for Weakly Labeled Semantic Video Object Segmentation, in *Proc. of the Asian Conf. on Computer Vision (ACCV) 2016*. Cited on page 153.

L. Wang, W. Ouyang, X. Wang, and H. Lu (2015a). Visual tracking with fully convolutional networks, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on page 29.

T. Wang, B. Han, and J. Collomosse (2014b). TouchCut: Fast image and video segmentation using single-touch interaction, *Computer Vision and Image Understanding (CVIU)*. Cited on pages 23 and 31.

W. Wang and J. Shen (2017). Super-Trajectory for Video Segmentation, *arXiv:1702.08634*. Cited on pages 28, 166, 173, and 181.

W. Wang, J. Shen, and F. M. Porikli (2015b). Saliency-aware geodesic video object segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 26.

X. Wang and I. Davidson (2010). Flexible constrained spectral clustering., in *Proc. of the International Conf. on Knowledge Discovery and Data Mining (KDD) 2010*. Cited on page 99.

X. Wang and A. Gupta (2015). Unsupervised learning of visual representations using videos, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 166 and 195.

Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan (2017). Object Region Mining With Adversarial Erasing: A Simple Classification to Semantic Segmentation Approach, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on page 21.

Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, Y. Zhao, and S. Yan (2015). STC: A Simple to Complex Framework for Weakly-supervised Semantic Segmentation, *arXiv:1509.03150*. Cited on pages 19, 21, 74, and 92.

L. Wen, D. Du, Z. Lei, S. Z. Li, and M.-H. Yang (2015). JOTS: Joint online tracking and segmentation, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 27, 28, and 152.

Z. Wu, F. Li, R. Sukthankar, and J. M. Rehg (2015). Robust video segment proposals with painless occlusion handling, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 25 and 26.

F. Xiao and Y. J. Lee (2016). Track and segment: An iterative unsupervised approach for video object proposals, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on pages 23, 26, 143, 153, 173, and 189.

S. Xie and Z. Tu (2015). Holistically-Nested Edge Detection, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 16, 36, 38, 39, 40, 47, 59, 67, 131, and 190.

C. Xu and J. J. Corso (2012). Evaluation of Super-Voxel Methods for Early Video Processing, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2012*. Cited on page 97.

C. Xu, S. Whitt, and J. J. Corso (2013). Flattening supervoxel hierarchies by the uniform entropy slice, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2013*. Cited on pages 109 and 111.

C. Xu, C. Xiong, and J. Corso (2012). Streaming Hierarchical Video Segmentation, in *Proc. of the European Conf. on Computer Vision (ECCV) 2012*. Cited on pages 24, 100, and 113.

J. Xu, A. Schwing, and R. Urtasun (2015). Learning To Segment under Various Forms of Weak Supervision, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on pages 17, 19, 20, and 73.

L. Xu, W. Li, and D. Schuurmans (2009). Fast normalized cut with linear constraints., in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2009*. Cited on page 99.

N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang (2017). Deep GrabCut for Object Selection, in *Proc. of the British Machine Vision Conf. (BMVC) 2017*. Cited on page 22.

N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang (2016). Deep Interactive Object Selection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2016*. Cited on page 68.

K. Yamada, Y. Sugano, T. Okabe, Y. Sato, A. Sugimoto, and K. Hiraki (2010). Can saliency map models predict human egocentric visual attention?, in *Proc. of the Asian Conf. on Computer Vision (ACCV) 2010*. Cited on page 81.

S. Yi and V. Pavlovic (2015). Multi-Cue Structure Preserving MRF for Unconstrained Video Segmentation, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 15, 23, 125, 127, 137, and 139.

F. Yu and V. Koltun (2016). Multi-Scale Context Aggregation by Dilated Convolutions, in *Proc. of the International Conf. on Learning Representations (ICLR) 2016*. Cited on pages 18 and 19.

H. Yu, Y. Zhou, H. Qian, M. Xian, Y. Lin, D. Guo, K. Zheng, K. Abdelfatah, and S. Wang (2015). LooseCut: Interactive Image Segmentation with Loosely Bounded Boxes, *arXiv:1507.03060*. Cited on page 22.

J. J. Yu, A. W. Harley, and K. G. Derpanis (2016). Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness, *arXiv:1608.05842*. Cited on page 166.

S. X. Yu and J. Shi (2001). Grouping with Bias., in *Advances in Neural Information Processing Systems (NIPS) 2001*. Cited on page 99.

C. Zach, T. Pock, and H. Bischof (2007). A Duality Based Approach for Realtime TV-L1 Optical Flow, in *Proc. of the DAGM Symposium on Pattern Recognition (DAGM) 2007*. Cited on page 114.

S. Zagoruyko, A. Lerer, T. Lin, P. O. Pinheiro, S. Gross, S. Chintala, and P. Dollár (2016). A MultiPath Network for Object Detection, in *Proc. of the British Machine Vision Conf. (BMVC) 2016*. Cited on page 22.

M. D. Zeiler and R. Fergus (2014). Visualizing and Understanding Convolutional Networks, in *Proc. of the European Conf. on Computer Vision (ECCV) 2014*. Cited on page 75.

M. D. Zeiler, G. W. Taylor, and R. Fergus (2011). Adaptive deconvolutional networks for mid and high level feature learning, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2011*. Cited on page 18.

D. Zhang, O. Javed, and M. Shah (2013). Video Object Segmentation through Spatially Accurate and Temporally Dense Extraction of Primary Object Regions, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2013*. Cited on pages 26 and 109.

J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff (2016). Top-Down Neural Attention by Excitation Backprop, in *Proc. of the European Conf. on Computer Vision (ECCV) 2016*. Cited on pages 75, 76, and 80.

J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Měch (2015a). Minimum Barrier Salient Object Detection at 80 FPS, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on page 82.

R. Zhang, P. Isola, and A. A. Efros (2017). Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2017*. Cited on page 195.

Y. Zhang, X. Chen, J. Li, C. Wang, and C. Xia (2015b). Semantic object segmentation via detection in weakly labeled video, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 153.

H. Zhao (2017). Some Promising Ideas about Multi-instance Video Segmentation, *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*. Cited on page 186.

H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia (2016). Pyramid Scene Parsing Network, *arXiv:1612.01105*. Cited on pages 19, 164, and 194.

R. Zhao, W. Ouyang, H. Li, and X. Wang (2015). Saliency detection by multi-context deep learning, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 87.

S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr (2015). Conditional Random Fields as Recurrent Neural Networks, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2015*. Cited on pages 19 and 62.

F. Zhong, X. Qin, Q. Peng, and X. Meng (2012). Discontinuity-aware video object cutout, *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH)*. Cited on page 31.

B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba (2016). Learning Deep Features for Discriminative Localization., *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Cited on pages 75, 76, 77, 87, and 191.

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros (2017a). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, in *Proc. IEEE International Conf. on Computer Vision (ICCV) 2017*. Cited on page 195.

Y. Zhu, Z. Lan, S. Newsam, and A. G. Hauptmann (2017b). Guided Optical Flow Learning, *arXiv:1702.02295*. Cited on page 166.

Y. Zhu, R. Urtasun, R. Salakhutdinov, and S. Fidler (2015). segDeepM: Exploiting Segmentation and Context in Deep Neural Networks for Object Detection, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Cited on page 35.

PUBLICATIONS

[8] *Lucid Data Dreaming for Multiple Object Tracking.*
Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele.
arXiv: 1703.09554, 2017.

[7] *Learning Video Object Segmentation from Static Images.*
Anna Khoreva*, Federico Perazzi*, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung.
In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (**CVPR**), 2017.

[6] *Simple Does It: Weakly Supervised Instance and Semantic Segmentation.*
Anna Khoreva, Rodrigo Benenson, Jan Hosang, and Bernt Schiele.
In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (**CVPR**), 2017.

[5] *Exploiting Saliency for Object Segmentation from Image Level Labels.*
Seong Joon Oh, Rodrigo Benenson, Anna Khoreva, Zeynep Akata, Mario Fritz, and Bernt Schiele.
In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (**CVPR**), 2017.

[4] *Weakly Supervised Object Boundaries .*
Anna Khoreva, Rodrigo Benenson, Mohamed Omran, and Bernt Schiele.
In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (**CVPR**), 2016.

[3] *Improved Image Boundaries for Better Video Segmentation.*
Anna Khoreva, Rodrigo Benenson, Fabio Galasso, Matthias Hein, and Bernt Schiele.
In 2nd International Workshop on Video Segmentation in conjuction with **ECCV** 2016, published in Computer Vision – ECCV 2016 Workshops, 2016 Proceedings, Part III.

[2] *Classifier Based Graph Construction for Video Segmentation.*
Anna Khoreva, Fabio Galasso, Matthias Hein, and Bernt Schiele.
In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (**CVPR**), 2015.

[1] *Learning Must-Link Constraints for Video Segmentation Based on Spectral Clustering.*
Anna Khoreva, Fabio Galasso, Matthias Hein, and Bernt Schiele.
In Proc. German Conf. on Pattern Recognition (**GCPR**), 2014.