Image Manipulation against Learned Models Privacy and Security Implications

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

by
Seong Joon Oh, M.Sc.

Saarbrücken
2018

Day of Colloquium	6th of August, 2018
Dean of the Faculty	Univ.-Prof. Dr. Sebastian Hack Saarland University, Germany
Examination Committee	
Chair	Prof. Dr. Antonio Krüger
Reviewer, Advisor	Prof. Dr. Bernt Schiele
Reviewer, Co-Advisor	Dr. Mario Fritz
Reviewer	Prof. Vitaly Shmatikov
Reviewer	Prof. Serge Belongie
Academic Assistant	Dr. Michael Xuelin Huang

ABSTRACT

Machine learning is transforming the world. Its application areas span privacy sensitive and security critical tasks such as human identification and self-driving cars. These applications raise privacy and security related questions that are not fully understood or answered yet: Can automatic person recognisers identify people in photos even when their faces are blurred? How easy is it to find an adversarial input for a self-driving car that makes it drive off the road?

This thesis contributes one of the first steps towards a better understanding of such concerns. We observe that many privacy and security critical scenarios for learned models involve input data manipulation: users obfuscate their identity by blurring their faces and adversaries inject imperceptible perturbations to the input signal. We introduce a *data manipulator framework* as a tool for collectively describing and analysing privacy and security relevant scenarios involving learned models. A *data manipulator* introduces a shift in data distribution for achieving privacy or security related goals, and feeds the transformed input to the target model. This framework provides a common perspective on the studies presented in the thesis.

We begin the studies from the user's privacy point of view. We analyse the efficacy of common obfuscation methods like face blurring, and show that they are surprisingly ineffective against state of the art person recognition systems. We then propose alternatives based on head inpainting and adversarial examples. By studying the user privacy, we also study the dual problem: model security. In model security perspective, a model ought to be robust and reliable against small amounts of data manipulation. In both cases, data are manipulated with the goal of changing the target model prediction. User privacy and model security problems can be described with the same objective.

We then study the *knowledge* aspect of the data manipulation problem. The more one knows about the target model, the more effective manipulations one can craft. We propose a game theoretic manipulation framework to systematically represent the knowledge level on the target model and derive privacy and security guarantees. We then discuss ways to *increase knowledge* about a black-box model by only querying it, deriving implications that are relevant to both privacy and security perspectives.

ZUSAMMENFASSUNG

Maschinelles Lernen verändert die Welt. Die Anwendungsbereiche umfassen datenschutzrelevante und sicherheitskritische Aufgaben wie die Personenidentifikation und selbstfahrende Autos. Diese Anwendungen werfen Fragen zum Datenschutz und zur Sicherheit auf, die noch nicht vollständig verstanden oder beantwortet sind: Können Personen auf Fotos durch automatische Personenidentifikation erkannt werden, selbst wenn ihre Gesichter verschwommen sind? Wie leicht ist es, eine feindliche Eingabe für ein selbstfahrendes Auto zu finden, die es von der Straße drängt?

Diese Arbeit trägt zu einem der ersten Schritte bei, um solche Probleme besser zu verstehen. Wir beobachten, daß viele datenschutz- und sicherheitskritische Szenarien für gelernte Modelle die Manipulation von Eingabedaten beinhalten: Benutzer verschleiern ihre Identität, indem sie ihre Gesichter unkenntlich machen, und Widersacher fügen dem Eingabesignal unmerkliche Störungen hinzu. Wir stellen ein Datenmanipulator-System als Werkzeug zur gemeinsamen Beschreibung und Analyse von datenschutz- und sicherheitsrelevanten Szenarien mit erlernten Modellen vor. Ein Datenmanipulator führt eine Verschiebung in der Datenverteilung ein, um Ziele bezüglich des Datenschutzes oder der Sicherheit zu erreichen, und leitet die transformierten Eingaben dem Zielmodell zu. Dieses System bietet eine gemeinsame Perspektive auf die in der Arbeit vorgestellten Studien.

Wir beginnen mit den Studien aus Sicht des Datenschutzes für den Benutzer. Wir analysieren die Wirksamkeit gängiger Verschleierungstechniken wie Gesichtsunschärfe und zeigen, daß sie überraschenderweise gegen moderne Personenerkennungssysteme unwirksam sind. Wir schlagen dann Alternativen vor, die auf Einfärben des Kopfes und Beispielen für feindliche Eingaben basieren. Durch das Studium des Benutzerdatenschutzes untersuchen wir auch das duale Problem: Modellsicherheit. Aus Sicht der Modellsicherheit sollte ein Modell robust und zuverlässig gegenüber kleinen Datenmanipulationen sein. In beiden Fällen werden Daten manipuliert mit dem Ziel, die Zielmodellvorhersage zu ändern. Probleme hinsichtlich des Benutzerdatenschutzes und der Modellsicherheit können mit demselben Ziel beschrieben werden.

Abschließend untersuchen wir den Wissens-Aspekt des Datenmanipulationsproblems. Je mehr man über das Zielmodell weiß, desto effektiver können Manipulationen durchgeführt werden. Wir schlagen ein spieltheoretisches Manipulationssystem vor, um das Wissensniveau auf dem Zielmodell systematisch darzustellen und Datenschutz- und Sicherheitsgarantien abzuleiten. Wir diskutieren dann Wege zur Wissenserweiterung über ein Black-Box-Modell, indem wir Ergebnisse von Anfragen dazu nutzen, Implikationen abzuleiten, die sowohl aus Sicht des Datenschutzes als auch der Sicherheit relevant sind.

CONTENTS

1 **Introduction** 1

1.1 Data manipulation framework 2

1.2 Outline of the thesis 6

2 **Related work** 9

2.1 Machine learning with non-iid data 9

2.2 Person recognition 10

2.3 User privacy over visual data 13

2.4 Machine learning security 14

2.5 Representing target uncertainty 17

2.6 Black-box models 18

I **Privacy Analysis in Visual Data** 21

3 **Person Recognition in Personal Photo Collections** 22

3.1 Introduction 22

3.2 PIPA Dataset 23

3.3 Task and Experimental Setup 25

3.4 Cues for recognition 28

3.5 PIPA test set results and comparison 37

3.6 Analysis 40

3.7 Open-world recognition 46

3.8 Conclusion 48

4 **Faceless Person Recognition in Social Media Photos** 51

4.1 Introduction 51

4.2 Privacy scenarios and setup 52

4.3 Faceless Recognition System 55

4.4 Test set results & analysis 63

4.5 Discussion & Conclusion 66

II **Privacy Solution in Visual Data** 67

5 **Head Inpainting for Privacy Protection** 68

5.1 Introduction 68

5.2 Head inpainting framework 70

5.3 Experiments 74

5.4 Conclusion 81

6 I-Pic Image Capture Framework 83
6.1 Introduction 83
6.2 Online Survey 86
6.3 I-Pic Architecture 89
6.4 I-Pic Design 91
6.5 Evaluation 97
6.6 Conclusion 105

III Knowledge on Target Model 107

7 Adversarial Perturbation for Privacy Protection 108
7.1 Introduction 108
7.2 User-Recogniser Game 109
7.3 Adversarial Image Perturbation Strategies 113
7.4 Empirical Studies 116
7.5 Discussion & Conclusion 122
7.6 Additional Results 122

8 Reverse-Engineering Black-Box Neural Networks 129
8.1 Introduction 129
8.2 Metamodels 130
8.3 Reverse-Engineering Black-Box MNIST Digit Classifiers 136
8.4 Reverse-Engineering and Attacking ImageNet Classifiers 145
8.5 Conclusion 148

9 Image Label Supervised Semantic Segmentation 149
9.1 Introduction 149
9.2 Guided Segmentation architecture 153
9.3 Finding good seeds 154
9.4 Finding the object extent 158
9.5 Experiments 163
9.6 Conclusion 169

10 Conclusion and future perspectives 171
10.1 Key insights and conclusions 171
10.2 Future perspectives 172

List of Figures 175

List of Tables 177

Bibliography 179

1 INTRODUCTION

Machine learning is transforming the world. With growing amounts of web data, better computing power, and more effective learning algorithms, machine learning now enables the automatic execution of a diverse array of tasks, from ones with perceptual (e.g. object detection (Ren *et al.*, 2015)) and artistic elements (e.g. style transfer (Johnson *et al.*, 2016)) to ones that are privacy sensitive (e.g. face recognition (Schroff *et al.*, 2015)) or security critical (e.g. automatic convenience store checkout[1] and self-driving cars (Bojarski *et al.*, 2016)). In particular, applications in privacy and security relevant tasks leave us wondering if we fully understand their implications: Can we still opt out of face recognition through face blurring? Are self-driving cars as reliable as humans in all conditions? Today, not even machine learning researchers have satisfactory answers to those questions.

In many security- and privacy-relevant scenarios, input data manipulation comes into play. Suppose that a user blurs her infant son's face from a photo she is about to post on a social network, hoping to protect his privacy. She is manipulating the test data with the goal of protecting her son's privacy against face recognition systems. From the security perspective, consider a terrorist who puts a carefully designed adversarial pattern on a signpost that will guide self-driving cars to crash into pedestrians. In both cases, deployed models face an intelligent entity who intentionally introduces a shift in the test data distribution to induce a desired outcome. This thesis focuses on the security and privacy implications of such data manipulations against learned models.

The primary focus of machine learning research has been under the *iid (independent, identically distributed)* data assumption, where in particular the training and test data arise from the same distribution. This assumption implies for example that a self-driving car will always encounter the road and weather conditions that it has been trained on. We have seen staggering advances in theory (e.g. Statistical Learning Theory) and empirical techniques (e.g. deep neural networks) in the iid setup during the last decade.

The privacy and security implications and guarantees under security- and privacy-relevant scenarios lack sufficient research compared to their urgency. There exists ongoing research on *adversarial examples* - small input perturbations that completely fail learned models - and corresponding defence measures. However, still far more investigation is needed to apply machine learning on security critical tasks with absolute confidence. On the privacy side, there has been only scant prior research on identity obfuscation (obscuring) techniques on visual data by the time we started our investigation. See the related work chapter (Chapter 2) for a more in-depth discussion.

[1] https://en.wikipedia.org/wiki/Amazon_Go

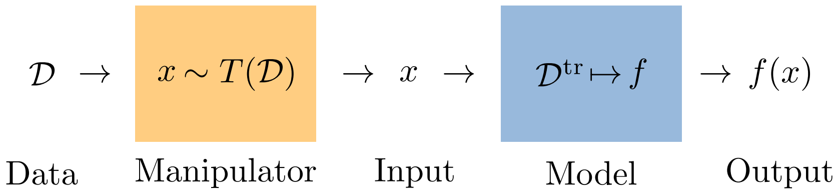


Figure 1.1: Overview of the privacy and security framework for learned models.

This thesis contributes to a more complete picture of the *privacy and security consequences of visual data manipulation by users*. In §1.1, we establish the data manipulation framework as a common language for formalising and describing privacy and security relevant scenarios. Under this framework, we start our investigation on user privacy in personal photo collections with respect to the latest advances in computer vision (Part I). In Part II, we propose novel data manipulation schemes (obfuscation) that provide both good identity protection and image naturalness. In Part III, we remark that the efficacy of data manipulation depends highly upon the manipulator's *knowledge* on the target model. We extend the data manipulation framework into a game theoretic one, where the model is underspecified, and analyse the utility guarantees for both the manipulator and the model. We also show that the manipulator can drastically increase its knowledge on the target model only through a sequence of queries. Note that discussion in Part III applies to both security and privacy. A more detailed outline of contributions per chapter is in §1.2.

Data manipulation is a double edged sword. Depending on the intent, the outcome may be good or bad for the society - this thesis does not make an ethical resolution on this point. We believe, however, that there is practically no way to prevent users from manipulating data altogether, and there must be a better understanding of both the threats and opportunities that it poses. This thesis contributes a scientific investigation towards both directions.

1.1 DATA MANIPULATION FRAMEWORK

We introduce the *data manipulation framework* in this section. The framework provides a common perspective to the chapters of the thesis. See Figure 1.1 for an overview. In our framework, the data manipulator is an agent with *goal*, *leverage*, and certain level of *knowledge* on the target model f. She transforms the *test data distribution* $\mathcal{D}$ into $T(\mathcal{D})$ according to her goal. We remark that $\mathcal{D}$ can be as small as a single image and as large as her entire photo album. An input x is then sampled from $T(\mathcal{D})$ and then fed to the model f. The model, having been trained on a certain *training distribution* $\mathcal{D}^{\mathbf{tr}}$, takes the input and outputs $f(x)$.

Depending on the manipulator's goal and knowledge of f, multiple privacy and security relevant scenarios arise. In this section, we walk through specific instances of this framework, linking to different chapters of the thesis.

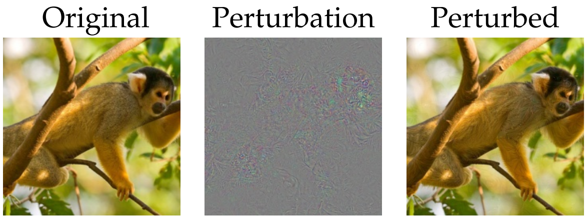


Figure 1.2: Adversarial examples from Chapter 8 (Oh *et al.*, 2018). The perturbation is nearly imperceptible, yet completely fools an image classifier.

1.1.1 Example: fooling machines

We consider the scenario where the manipulator manipulates the data to make the model behave abnormally. Data transformed in such a way are called *adversarial examples*. For example, if the target model is a classifier, the manipulator's goal could be to induce wrong label predictions by the classifier. Adversarial examples have been one of the most researched topic in the recent years, due to the seriousness of their implications on security critical applications (self-driving cars for example). We examine how this scenario can be cast into our framework.

The manipulator's goal is twofold: (1) wrong label prediction and (2) stealthy manipulation. If the manipulation is too obvious (e.g. filling in the entire image with black pixels), then the model may simply reject this input. Thus, in general, there exists a trade-off between the two factors. As a summary measure of the two goals, we define a *utility* function for the manipulator $U = U(T)$ that depends on the choice of the manipulation algorithm T. The stealthy manipulation goal is often measured in terms of the L_2 distance from the original input vector, as a proxy to "obviousness". The manipulator then chooses her transformation function according to the optimisation problem

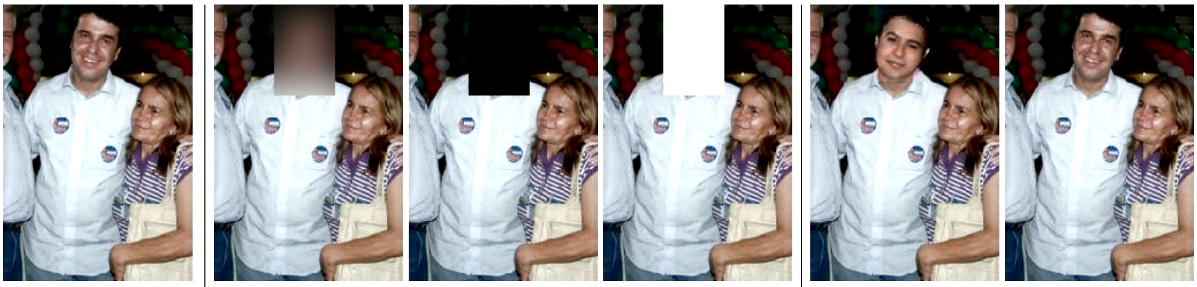
$$\max_{T \in \mathcal{T}} U(T) \tag{1.1}$$

where $\mathcal{T}$ is the leverage space for the manipulator. A specific instance of the adversarial example generation on a single image x with label y against a classifier f that returns a per-class probability vector is

$$\min_{\delta \in \mathcal{T}} \; \log f^y(x+\delta) + \lambda ||\delta||_2 \tag{1.2}$$

where $\mathcal{T}$ is the set of additive perturbations that ensures that the perturbed image $x + \delta$ is a valid image (within the cube $[0,1]^D$); $\lambda \geq 0$ is a scalar for determining the trade-off between the attack effectiveness (negative log likelihood in the first term) and the "obviousness" of the attack (second term). Note that $f^y(x+\delta)$ is the probability for class y that is dependent on the probability of the other classes (e.g. via softmax); decreasing $f^y(\cdot)$ increases $f^{y'}(\cdot)$ for $y' \neq y$.

The needed amount of perturbation to mislead a neural network classifier is surprisingly small - nearly imperceptible to human eyes (Figure 1.2). This observation



Original	Face blurring Ch.4	Blacking out Ch.4	Whitening out Ch.4	Inpainting Ch.5	Adversarial Ch.7
		Existing methods		Proposed methods	
Natural?	✗	✗	✗	✓	✓✓
Effective?	✗	✗	✗	✓	✓✓
Target generic?	✓	✓	✓	✓	✗

Figure 1.3: Identity obfuscation methods.

has spurred a lot of research on adversarial examples afterwards. See §2.4 for an in-depth discussion of prior work.

1.1.2 Example: identity obfuscation

We present an example where data manipulation can help users protect their privacy by letting them avoid automatic human identification. Previous researches have focused on the negative implications of data manipulation; this thesis is one of the first works in the field advocating an active use of the data manipulation for user privacy protection. The goal of the manipulator is twofold: (1) Avoid recognition and (2) maintain input naturalness. Under the social media photo sharing scenario, the second constraint prevents the manipulator from removing the entire image pixels and harming the image usefulness. The manipulation problem for privacy protection is defined similarly as in Equation 1.1.

There are two types of data manipulation schemes for identity protection: changing appearances themselves (e.g. hair colour, wearing sunglasses) or obfuscating identity sensitive regions post hoc (e.g. blurring or blacking-out faces, see Figure 1.3)? We quantify the efficacy of both schemes against our person recogniser in Part I. We conclude that they do not work as well as one might expect.

As better alternatives, this thesis proposes two novel obfuscation schemes, one based on head inpainting (Chapter 5) and the other one based on adversarial examples (Chapter 7). Figure 1.3 shows how they compare to face blurring and blacking-out in terms of obfuscation performance and image naturalness. The novel techniques clearly improve over existing ones.

1.1.3 Knowledge on the model

So far we did not take the manipulator's knowledge on the model into account. In practice, the manipulator's knowledge on f is often limited, and its utility depends on the level of knowledge on f: $U = U(T, f)$. For example, to generate adversarial examples, one computes model gradients on the input $\nabla_x f(x)$ for best performance (Goodfellow *et al.*, 2015); these gradients are not accessible for many deployed models. The adversarial example based identity obfuscation scheme (Chapter 7) achieves great image naturalness and obfuscation effectiveness, but comes at the cost of requiring gradient knowledge on the target (we refer to this as "non-target generic" in Figure 1.3). Since the manipulation performance depends highly on this knowledge, it is crucial to correctly reflect the knowledge level in choosing the manipulation scheme T.

Representing uncertainty in one's knowledge for control and decision problems has been an important subject of study in many academic fields including economics, robotics, operational research, optimization theory, and physics. In particular, to deal with parameter uncertainty in optimisation problems, robust (or stochastic) Optimisation (Ben-Tal *et al.*, 2009; Prékopa, 1995) has emerged. For Optimal Control Theory (Seierstad and Sydsaeter, 1986), robust control theory (Hansen and Sargent, 2001) has been studied as a robust surrogate. If one treats the uncertainty as a result of other agents' intelligent activities, game theory (Nash *et al.*, 1950) comes into play. In the following, we make connections between the treatment of model uncertainty under our framework and various fields of study listed above.

There are largely two ways of representing the uncertainty on f. The first method is to put a prior distribution over $f \sim \mathcal{D}_{\mathcal{F}}$. The manipulator's problem in this case is to optimise the marginalised utility:

$$\max_{T \in \mathcal{T}} \mathop{\mathbb{E}}_{f \sim \mathcal{D}_{\mathcal{F}}} \left[U(T, f) \right]. \tag{1.3}$$

There are problems with this scheme. First of all, it is often difficult to define and make sense of the prior distribution. Even worse, in real life the model may as well be a dynamic, intelligent entity with the ability to change f with the goal to thwart the manipulator's goal.

A second way of modelling the uncertainty is to derive the *worst-case* bound. Instead of setting a prior distribution for f, we set a candidate space for f ($f \in \mathcal{F}$), and assume that the model side always chooses the model that minimises the manipulator's utility:

$$\max_{T \in \mathcal{T}} \min_{f \in \mathcal{F}} U(T, f). \tag{1.4}$$

This optimisation problem gives a guarantee on the utility that is independent of the choice of model $f \in \mathcal{F}$, and the corresponding T for achieving that estimate. This framework successfully addresses the two problems of the marginalisation scheme (Equation 1.3): (1) we do not need to know the prior distribution and (2) the model may be adversarially chosen. In this framework, we can also represent fine-grained levels of knowledge, controlling the size of the candidate space $\mathcal{F}$.

Equation 1.4 is an instance of robust optimisation (Ben-Tal *et al.*, 2009). Equation 1.3 is a special case of stochastic optimisation (Prékopa, 1995) whose more general formulation constrains the *set of probability measures* over the models:

$$\max_{T\in\mathcal{T}} \min_{\mathcal{D}\in\mathbb{F}} \mathop{\mathbb{E}}_{f\sim\mathcal{D}} \left[U(T,f)\right]. \tag{1.5}$$

where $\mathbb{F}$ is a set of probability measures $\mathcal{D}$ over the models. Note that setting $|\mathbb{F}| = 1$ gives Equation 1.3 and setting $\mathbb{F}$ as the set of all single-mass (dirac-delta) measures over $\mathcal{F}$ gives Equation 1.4. Thus, Equation 1.5 is a generalisation of both formulations. However, Equation 1.5 results in a lack of interpretability (what does it mean to constrain a set of probability measures over models?) and in computationally intractable optimisation in general (Ben-Tal *et al.*, 2009).

We have adopted the robust optimisation form (Equation 1.4) in Chapter 7. We have considered both the manipulator and model as intelligent agents, so this is furthermore an instance of game theory. We have obtained privacy guarantees for the data manipulator as a social media user. However, the same framework can yield the security guarantees for the model against a malicious data manipulator (dual problem). There is an increasing body of work on applying this Game theoretic view on defences against adversarial examples, attempting to compute security guarantees of learned models against input perturbations (after publication of materials in Chapter 7, (Oh *et al.*, 2017c)).

1.1.4 Gaining more knowledge

In general, narrowing down the candidate space $\mathcal{F}$ will increase the optimal utility in Equation 1.4; i.e. more knowledge helps. We show in Chapter 8 that it is possible to dramatically increase one's knowledge on various hyperparameters of a black-box model through a sequence of queries (black box access). We verify that hyperparameters like the type of non-linear activation, training algorithm, and training dataset can be reliably reverse-engineered by interpreting the output patterns with respect to a sequence of (perhaps carefully designed) query inputs (`kennen` methods).

This implies that granting black box accesses to users may expose much more internal information than previously believed. In Chapter 8, we show further that the exposed internals elevate black-box models' susceptibility to adversarial examples. This raises new concerns on the model security that have not been addressed before.

1.2 OUTLINE OF THE THESIS

We provide an outline of the chapters according to the three parts of the thesis.

Part I: Privacy Analysis in Visual Data. We investigate identifiability of humans in visual data with the recent advances in machine learning and computer vision in mind. In Chapter 3, based on the conference paper "Person Recognition in Personal Photo Collections" (Oh *et al.*, 2015, 2017a), we build a state of the art person recognition system based on deep neural networks (`naeil`) and present an in-depth analysis of its stability against domain shifts. Specifically, we consider time shifts (change of clothing, events, activities, and time of the day) and head viewpoint changes (frontal versus back-view). We show that `naeil` is robust to those shifts, and that humans are identifiable in those personal photos better than previously believed (e.g. you can be recognised from the back-view). In Chapter 4, based on the conference paper "Faceless Person Recognition; Privacy Implications in Social Media" (Oh *et al.*, 2016), we consider more active means of identity obfuscation including face blurring or blacking out. We empirically argue that those common techniques are not sufficiently effective against state of the art recognisers which can easily make use of context information and are highly adaptive. These considerations have not been studied extensively in the computer vision or privacy domains before.

Part II: Privacy Solution in Visual Data. We present our technical contributions on identity obfuscation methods. Existing obfuscation methods are not only ineffective against state of the art recognition systems but also unnatural (Chapter 4). We focus on both obfuscation success rate and naturalness. In Chapter 5, based on the conference paper "Natural and Effective Obfuscation by Head Inpainting" (Sun *et al.*, 2018), we introduce a head inpainting method based on a generative adversarial network (GAN) that seamlessly replaces faces with plausible faces of non-existent identities. Chapter 7 is included in Part III, but also proposes an adversarial example based obfuscation technique. Unlike the inpainting method, adversarial examples make unnoticeable changes to human eyes to completely disable recognition by certain target models. However, they require specific knowledge on the target model for good performance. We introduce a Game theoretic framework to obtain privacy guarantees as a function of the knowledge on the target model (§1.1.3). Chapter 6 is an interlude chapter based on the conference paper "I-Pic: A Platform for Privacy-Compliant Image Capture" (Aditya *et al.*, 2016). We introduce the I-Pic image capture system that protects bystanders' privacy at capture stage.

Part III: Knowledge on Target Model. This part contributes to the formalisation of the manipulator's knowledge on the target model. Chapter 7, based on the conference paper "Adversarial Image Perturbation for Privacy Protection – A Game Theory Perspective" (Oh *et al.*, 2017c), introduces a Game theoretic framework to represent the lack of knowledge on the target, and corresponding utility guarantees for the participating entities. In Chapter 8, based on the conference paper "Towards reverse-engineering black-box neural networks" (Oh *et al.*, 2018), we demonstrate a black-box revealing technique that we call `kennen` which can reveal a variety of inner details of a model - architecture, training algorithm, and training data - only from a sequence of black-box accesses. We show further that this increased knowledge can aggravate the susceptibility of the exposed model to adversarial examples. Finally, Chapter 9 is another interlude chapter based on the conference paper "Exploiting Saliency for Object Segmentation from Image Level Labels" (Oh *et al.*, 2017b). Using techniques to extract knowledge on object locations from image classifiers, we train an image segmentation network. This results in much cheaper supervision (only image tags) than the full supervision (pixel wise labels) without a severe cost in performance.

1.2.1 Summary of participation

Each main chapter of the thesis is based on a conference paper. Seong Joon Oh has participated in each chapter (paper) with varying degrees of contribution. We provide a summary here.

First-author participation. Seong Joon Oh has participated in Chapters 3, 4, 7, 8, and 9 as the first author. Under the supervision of the advisors and in close collaboration with colleagues, he has conducted experiments, written manuscripts, and presented the materials in conferences and talks. With the exception of §9.4, in which the weakly supervised saliency network and related experiments are contributed by Dr Anna Khoreva (co-author), all sections are primarily and materially contributed by Seong Joon Oh.

Co-author participation. As a co-author, Seong Joon Oh has partially contributed to Chapters 6 and 5. In Chapter 5, he has evaluated the proposed inpainting-based identity obfuscation technique against his recognition system `naeil` (Chapter 3). He has also substantially contributed to the writing of this chapter. In Chapter 6, he has implemented the person recognition module for the privacy-preserving image capture platform I-Pic.

RELATED WORK 2

Before presenting the main work, we recap the prior as well as ongoing work in the interdisciplinary field of machine learning, computer vision, security, and privacy, focusing on the data manipulation aspect. We start off with a brief overview of previous studies in machine learning with non-iid data in general (§2.1). Since our study on user privacy makes extensive use of our state of the art person recogniser (`naeil`, Chapter 3), we discuss relation to other research on human identification in §2.2. We then move on to previous research on user privacy over visual data (§2.3) and machine learning security against data manipulation (§2.4). Prior work on the knowledge aspect of general manipulation problems is then discussed in §2.5. Finally in §2.6, we present related work on the increasing manipulator's knowledge by extracting internal information from black-box models.

2.1 MACHINE LEARNING WITH NON-IID DATA

Machine learning has thrived on the iid (independent, identically distributed) data assumption. Even in the iid setup, models need to generalise from finite training data samples to the population distribution. This can already be a challenging task. Many theoretical and empirical research efforts have been put into understanding and enhancing the performance in the iid setup. From the theory side, statistical learning theory gives stochastic guarantees on the iid generalisability (Uniform Convergence Theory, (Vapnik and Chervonenkis, 2015)). On the empirical side, deep learning algorithms have been the most successful ones, attaining the top performances in many vision benchmarks (Deng *et al.*, 2009; Krizhevsky *et al.*, 2012; Huang *et al.*, 2007; Everingham *et al.*, 2015; He *et al.*, 2016).

Non-iid setup is less explored than the iid setup, but there has been a lot of effort on understanding and improving machine learning performances under this setup. Darrell *et al.* (2015) provide an overview of the existing and future directions of machine learning research under this setup. We focus in particular on the cases where the training and testing data arise from different distributions (*domain shift*). We further make a distinction whether the domain shift arises with an intention or not. The thesis is primarily focused on the intelligent, purposeful manipulation of the testing data.

2.1.1 Naturally arising domain shift

We first review prior research on machine learning under domain shift that arises naturally, without an intelligent behaviour behind the scene. The domain shift could be a result of different modalities (e.g. image versus sound), different sensors (e.g. DSLR versus egocentric cameras), different processing (e.g. raw versus jpeg compressed images), or non-stationary data generation (e.g. hair colour changes in personal photos).

Machine learning techniques for dealing with domain shift are often referred to as *domain adaptation* or *domain transfer* methods. On the theory side, we have the seminal work by Ben-David *et al.* (2010) that has bounded the target domain loss by the source domain loss plus some domain difference term. There is a much richer body of work on empirical approaches. After the proposal of the milestone benchmark for evaluating domain adaptation (Saenko *et al.*, 2010) (across photos from different domains – Amazon, DSLR, and webcam images), many empirical approaches have appeared. Most recent deep learning based approaches advocate the *adversarial domain adaptation* technique (Long *et al.*, 2015b; Ganin and Lempitsky, 2015; Ganin *et al.*, 2016) that guides the feature space to be indiscriminative with respect to the domain through an adversarial loss.

In Chapter 3, after introducing `naeil`, our state of the art person recogniser, we measure its performance under time and viewpoint domain shifts. We show that `naeil` generalises well across these domain gaps thanks to the combination of time-stable cues (e.g. face) and viewpoint stable cues (e.g. body and scene).

2.1.2 Intentional domain shift

While the domain shift can arise naturally, it can also be a result of deliberate manipulation. This thesis focuses on this type of domain shift. We consider data manipulation with an intent to protect privacy against person recognition models as well as with a malicious intent to fail a deployed model or to expose its internal hyperparameters. We will discuss relevant prior works in §2.3 and §2.4.

2.2 PERSON RECOGNITION

In the thesis, we build our own person recognition system (`naeil`, Chapter 3) that reliably recognises people in personal photos, where they pose naturally and significant time and viewpoint shifts may take place. We refer to human identification in such a setup as *person recognition in personal photos*. We review related research in computer vision on human identification, and discuss how the settings differ and how our method `naeil` is different from others developed under the same setting.

2.2.1 Data type and cues

Humans in visual media can be recognised from a variety of cues. The computer vision and biometrics communities usually focus on face and body cues. On the other hand, person recognition in personal photos should not solely rely on these cues but combine evidence from face, body, and perhaps other contextual information for best performance. We discuss some relevant literature according to the cues considered.

Face. Face has been the traditional focus of human identification research. The Labelled Faces in the Wild (LFW) dataset (Huang *et al.*, 2007) has been a great testbed for a host of work on face identification and verification *outside the lab setting*. Attributing to the deep features trained on large scale face databases (Taigman *et al.*, 2014; Sun *et al.*, 2015; Schroff *et al.*, 2015), the performance has nearly saturated in this benchmark. However, LFW is not representative for personal photos taken in daily lives: the data consists mainly of unoccluded frontal faces and has a bias towards public figures. Indeed, more recent benchmarks have introduced more difficult types of data. IARPA Janus Benchmark A (IJB-A, (Klare *et al.*, 2015)) includes faces with profile viewpoints, but is still limited to public figures.

Body. The body region has been another important cue for human identification. Pedestrian re-identification (re-id) tackles the problem of matching pedestrians captured in different camera views. On standard benchmarks (VIPeR, Gray *et al.* (2007); CAVIAR, Cheng *et al.* (2011); CUHK, Li *et al.* (2012); Caltech Roadside Pedestrians, Hall and Perona (2015)), convnet architectures have led to great advances (Li *et al.*, 2014a; Yi *et al.*, 2014a; Hu *et al.*, 2014; Ahmed *et al.*, 2015; Cheng *et al.*, 2016; Xiao *et al.*, 2016; Varior *et al.*, 2016; Chen *et al.*, 2017b). However, typically the re-id benchmarks do not fully cover the human identification problem in person photos in three aspects: (1) subjects mostly appear in a standing pose, (2) resolution is low, and (3) matching is only evaluated across a short time span.

Person recognition in personal photos. Recognising people in natural poses with great variations in time and viewpoint requires fusing multiple cues. An early work towards this direction was done by Anguelov *et al.* (2007) who used both face and clothing cues for recognising people. A small-scale dedicated dataset for person recognition in personal photos was contributed by Gallagher and Chen (2008) – the "Gallagher collection person dataset". MegaFace (Kemelmacher-Shlizerman *et al.*, 2016; Nech and Kemelmacher-Shlizerman, 2017) is perhaps the largest open source face database over personal photos (Flickr). However, MegaFace does not contain any subject seen from the back. In this thesis, we make extensive use of the PIPA dataset (Zhang *et al.*, 2015b), a large scale ($\sim$40k images, $\sim$2k identities) dataset of Flickr personal photos, with diverse appearances and subjects with all viewpoints and occlusion levels. Heads are annotated with bounding boxes and with an identity tag. We describe PIPA in greater detail in §3.2. Recognition methods developed over this dataset are discussed in §2.2.3.

2.2.2 Recognition tasks

There exist multiple tasks related to person recognition (Gong *et al.*, 2014) differing mainly in the amount of training and testing data. Face and surveillance re-identification is most commonly done via verification: *given one reference image* *(gallery) and one test image (probe), do they show the same person?* (Huang *et al.*, 2007; Bedagkar-Gala and Shah, 2014). In this thesis, we consider two classification tasks:

- Closed world recognition: Given a single test image (probe), who is this person among the identities that are among the training identities (gallery)?

- Open world recognition: Given a single test image (probe), is this person among the training identities (gallery set)? If so, who? (Kemelmacher-Shlizerman *et al.*, 2016)

Other related tasks are, face clustering (Cui *et al.*, 2007; Schroff *et al.*, 2015), finding important people (Mathialagan *et al.*, 2015), or associating names in text to faces in images (Everingham *et al.*, 2006, 2009).

2.2.3 Person recognition in personal photos – methods

Prior to the introduction of the PIPA dataset (Zhang *et al.*, 2015b), only a few researchers have developed human identification systems that recognise people from both face and non-face cues in unrestricted personal photos. Examples include Gallagher and Chen (2008) and Anguelov *et al.* (2007), but those methods did not benefit from deep learning and massive training data that appeared afterwards and were tested only on small-scale benchmarks.

The PIPA dataset (Zhang *et al.*, 2015b) has enabled large-scale training and evaluation; related research field has bloomed afterwards. The first recognition method proposed was the so-called Pose Invariant Person Recognition method (`PIPER`, (Zhang *et al.*, 2015b)), obtaining promising results by combining a face recognition module (Taigman *et al.*, 2014), poselets (Bourdev and Malik, 2009), and convnet features trained on detected poselets (Krizhevsky *et al.*, 2012; Deng *et al.*, 2009).

Oh *et al.* (2015), the conference paper of the material in Chapter 3, has greatly simplified the recognition procedure (`naeil`), surpassing the performance of `PIPER` that uses more than 100 cues by using only head and body cues. In particular, `naeil` does not require the data-heavy DeepFace method or time-costly poselets. In Oh *et al.* (2015), (Chapter 3), we have augmented `naeil` with a face recognition module (`DeepID2+`, (Sun *et al.*, 2015)) to achieve new state of the art result as of 2018.

There have been many follow-up papers after we published Oh *et al.* (2015). Kumar *et al.* (2017) have improved the performance by normalising the body pose using pose estimation. Li *et al.* (2017) considered exploiting people co-occurrence statistics. Liu *et al.* (2017b) have proposed to train a person embedding in a metric space instead of training a classifier on a fixed set of identities, thereby making the model more adaptable to unseen identities. Some works have exploited the photo-album metadata, allowing the model to reason over different photos (Oh *et al.*, 2016; Li *et al.*, 2016a).

2.3 USER PRIVACY OVER VISUAL DATA

Privacy is one of the greatest interests of modern human beings (Holvast, 2008). With the technological advances, people have migrated substantial amount of private information in digital formats, often stored using cloud services. Along with technologies for securely storing such data, researchers have studied the level of privacy leakage due to private information shared in public (especially on social media) (Narayanan and Shmatikov, 2009, 2010; Zheleva and Getoor, 2009; Narayanan and Shmatikov, 2010; Mislove *et al.*, 2010). However, prior research has mostly focused on non-visual data (e.g. textual). There are works which consider the relationship between privacy and photo sharing activities such as Ahern *et al.* (2007) and Besmer and Richter Lipford (2010), yet they do not perform quantitative studies.

This thesis is one of the first to quantify privacy leakage in visual data. In particular, we measure the identifiability of people in personal photos in the context of the recent developments in automatic person recognisers (Chapters 3 and 4). In this section, we introduce related work that appeared before and after our work on user privacy in visual data.

2.3.1 Quantification of privacy leakage through visual media

Studies on quantifying privacy in visual data began only recently. Before us, Wilber *et al.* (2016) have quantified the decrease in face *detection* accuracy with respect to different types of obfuscation - e.g. blur, blacking-out, swirl, and dark spots, arguing that if face detection fails subsequent identification will inevitably fail. However, this argument overlooks the fact that the considered obfuscation patterns may be highly detectable themselves. We argue that it is important to directly analyse obfuscation performance against the identification system. Chapter 4 (Oh *et al.*, 2016) considers the *identification* problem with a recognition model *adapted* to obfuscation patterns. A few other works studied face recognition under blur (Gopalan *et al.*, 2012; Punnappurath *et al.*, 2015), but to the best of our knowledge, we are the first to consider person recognition under head obfuscation using an adaptive system that leverages full-body cues. A concurrent work (McPherson *et al.*, 2016) studies a similar problem, but did not take the adaptability of models into account. After the publication of our work, Orekondy *et al.* (2017) have studied the visual privacy problem from a broader perspective, covering not only person identity information but also other private information like credit card number and fingerprint.

2.3.2 Obfuscation methods

To avoid recognition either from other humans or machine systems, humans have manipulated data in certain ways. Wearing masks is one such way. Some governments have even decreed anti-mask laws in an attempt to preclude certain crimes[2]. At the

[2] https://www.nytimes.com/2017/04/26/us/protests-masks-laws.html

digital age, it is more common to manipulate the captured image - e.g. face blurring. While being used extensively (e.g. YouTube face blurring[3]), there is not so much quantitative work on their effectiveness, especially in the context of recent advances in machine learning that has enabled large-scale retrieval of private information. This thesis is one of the first work on measuring the identifiability of humans in visual data for a state of the art person recogniser, when various obfuscation patterns are applied. In this section, we review existing and ongoing work on the analysis of privacy in visual and non-visual media as well as protection techniques.

Some work from the vision community has developed obfuscation patterns for protecting private visual content. Hassan *et al.* (2017) have proposed to mask private image content via *cartooning*. Brkic *et al.* (2017) have generated full-person patches to overlay on top of person masks. Similarly, we propose an obfuscation technique based on *head inpainting* in Chapter 5 (Sun *et al.*, 2018). The key advantage of our approach is that unlike Brkic *et al.* (2017) who have generated persons with uniform poses independent of the context in fashion photographs, we inpaint heads that blend naturally into varied background and body poses in unrestricted personal photographs. Sharif *et al.* (2016) have proposed adversarial example based obfuscation techniques. Similarly, in Chapter 7 (Oh *et al.*, 2017c) we introduce novel obfuscation schemes based on adversarial examples. While Sharif *et al.* (2016) assume a full knowledge on the target recognition system, we have proposed a game theoretic framework to embrace certain amount of misspecification of the target model (more details in §2.5).

2.4 MACHINE LEARNING SECURITY

In this section we take a different point of view: we pose the data manipulator as a "bad guy" trying to undermine a model deployed for a security critical task. We include this point of view in the thesis in Part III. Like privacy, security and robustness of engineered systems have long been a topic of interest. The advent of successful machine learning models have created an interdisciplinary field of "machine learning security", in which the reliability and stability issues of learned models are studied. We review some work in this field that are relevant to the thesis – i.e. *adversarial examples* and corresponding defence techniques. For a wider overview of the field, see Papernot *et al.* (2018).

2.4.1 Adversarial examples

Adversarial examples are test-time attacks on machine learning models – the adversary introduces a "small" perturbation on the input to guide the model to behave in a certain way (often to fail on certain inputs). Adversarial examples against learned models were already being discussed as early as 2003. Lanckriet *et al.* (2003) proposed a generic minimax framework for improving robustness of a classifier against

[3] https://youtube.googleblog.com/2012/07/face-blurring-when-footage-requires.html

both naive and adversarial *feature* perturbations. We have seen much research on adversarial feature perturbations with the typical scenario from the spam filtering domain (Wittel and Wu, 2004; Lowd and Meek, 2005; Barreno *et al.*, 2006; Biggio *et al.*, 2008). See Cormack (2008) for an overview.

After the appearance of successful deep, end-to-end models (Krizhevsky *et al.*, 2012), it has been observed that the amount of perturbation needed to fool a trained convnet is nearly imperceptible to human observers (Szegedy *et al.*, 2014). This implies two important consequences: (1) an adversary can easily sneak in an adversarial input without being detected, and (2) the way machines perceive the world is very different from that of humans even if on benign test data their performances match humans'. For those two reasons, adversarial examples have attracted a great deal of attention and follow-up research. As of today, adversarial examples against neural networks are one of the most active area of research in machine learning and security communities.

Generation algorithms. There have been several key advances in adversarial example generation algorithms. The first work (Szegedy *et al.*, 2014) relied on an inefficient L-BFGS optimisation. Soon, efficient first-order algorithms have been proposed (Goodfellow *et al.*, 2015; Rozsa *et al.*, 2016; Moosavi-Dezfooli *et al.*, 2016; Kurakin *et al.*, 2017a; Carlini and Wagner, 2017b), which use the image gradient with respect to the adversarial goal to generate the perturbation. This thesis has contributed to the community by proposing a robust variant of DeepFool (Moosavi-Dezfooli *et al.*, 2016), `GAMAN` (Chapter 7, (Oh *et al.*, 2017c)). Recent trends involve learning to generate perturbations (Baluja and Fischer, 2018; Zhao *et al.*, 2018) and finding non-additive perturbations that are not regulated by the L_p metric (Xiao *et al.*, 2018; Zeng *et al.*, 2017; Kanbak *et al.*, 2018).

Other types of adversarial examples. Other papers have discussed the problem of generating a single adversarial example that could work for multiple input images ("Universal adversarial perturbations", (Moosavi-Dezfooli *et al.*, 2017)) or multiple target networks (Liu *et al.*, 2017a). Not only under the supervised learning setting but also under the reinforcement learning (RL) setting (Huang *et al.*, 2017b) were adversarial examples developed.

Robust adversarial examples. Naively generated adversarial examples are known to be fragile. Graese *et al.* (2016) argued that simple test time image processing, such as translation, Gaussian noise, blurring, and re-sizing can neutralise the adversarial effects; similarly, Lu *et al.* (2017b) have shown that the adversarial patterns do not survive physical printing. There are several attempts to re-strengthen the adversarial examples against those image-level defences. Sharif *et al.* (2016) have discovered sturdier adversarial patterns by combining the image gradients against a set of jittered versions of the input with a total variation regularisation. Athalye *et al.* (2018) have even 3D printed adversarial "objects" that result in wrong model classifications when captured with a camera in multiple viewpoints. In Chapter 7 (Oh *et al.*, 2017c), we also consider generating the perturbation against a set of jittered versions of the input to improve robustness to specific types of input jittering.

2.4.2 Defence against adversarial examples

Defence measures against adversarial examples has equally attracted interests in the research community. We identify five main approaches to defence.

Gradient masking. Motivated by the fact that most adversarial examples exploit the model gradient, several papers have suggested reducing the size of the input gradients of the deployed models (Papernot *et al.*, 2016b). This approach has some loopholes. First of all, one can generate adversarial examples without the gradient information (e.g. black-box attacks; see §2.6.2). Moreover, it has been found that slight modifications of the gradient computation can easily re-enable the attack (Carlini and Wagner, 2016).

Adversarial training. One can include adversarial examples during training to match the training distribution to the adversarial testing distribution. Goodfellow *et al.* (2015) have trained their model with relatively weak adversarial examples (FGSM) and have shown mediocre protection at test time. Stronger adversaries during training in general result in better robustness (Huang *et al.*, 2015). Madry *et al.* (2018) have added simple yet strong projected gradient descent (PGD) adversarial examples during training to reach a state of the art level defence. Kurakin *et al.* (2017b) discuss techniques for scaling up the adversarial training to ImageNet (Deng *et al.*, 2009) classifiers.

Detection & rejection. It is hypothesised that adversarial perturbations introduce certain detectable patterns on the input. If those patterns can be detected, then adversarial inputs may be filtered out before harming the model. Researchers have proposed to perform statistical testing (Grosse *et al.*, 2017) or feature projection (squeezing) (Xu *et al.*, 2017). Lu *et al.* (2017a) and Metzen *et al.* (2017) have trained dedicated detectors to filter out adversarial examples. However, Carlini and Wagner (2017a) have demonstrated counter-strategies against those detection based defences.

Defence by denoising. The denoising line of work aims to further modify the image to remove the adversarial effect. Das *et al.* (2017) and Dziugaite *et al.* (2016) have proposed to use JPEG compression over potential adversarial examples to remove adversarial patterns. However, it is in general easy to re-adapt adversarial examples again to those differentiable image processing steps (Chapter 7). On the other hand, Moosavi-Dezfooli *et al.* (2018) have proposed to divide the image into regions and denoise each region, a non-differentiable procedure.

Security guarantees. Using a game theoretical (or a robust optimisation) minimax formulation, multiple prior works have obtained robustness certificates for learned models, in the sense that certain data points within some L_p balls around training data points will not be able to fool the models. However, such a guarantee can only be obtained for a very small certificate (Hein and Andriushchenko, 2017) or after an NP-hard computation (Carlini *et al.*, 2018). Sinha *et al.* (2018) have proposed a polynomial time algorithm to obtain a certificate, but the certificate is in a less interpretable form.

2.4.3 Training time data manipulation

More commonly referred to as *poisoning attack*, training data manipulation and the corresponding defence measure (Steinhardt *et al.*, 2017) have been been investigated (Biggio *et al.*, 2012; Koh and Liang, 2017). This thesis focuses on the case where the manipulator only has access to the test inputs to the deployed models.

2.5 REPRESENTING TARGET UNCERTAINTY

In many application scenarios, deployed models hide internal details of the model, such as the architecture and training data. Thus, for example, gradient with respect to the input cannot readily be computed using an efficient backpropagation algorithm, leading to inefficient data manipulation against the model. It is important to precisely represent the level of knowledge for the manipulator about the model as it generally substantially influences the manipulation efficacy.

We first discuss what it means to "specify" a model. Mathematically, a function $f : X \to Y$ is fully specified if its domain X (and co-domain Y) is known and for every input $x \in X$ the corresponding output $f(x) \in Y$ is known. A model is essentially a function $f : X \to Y$, but is also implemented algorithmically, so the knowledge on the specific algorithmic implementation matters for computability and efficiency. For example, if an image classifier is implemented as a lookup table, it will be computationally prohibitive to compute the output as well as the gradients, while deep neural network implementations (i.e. composition of elementary subdifferentiable functions) with suitable hardware (e.g. GPUs) make it efficient to compute the model outputs and gradients. Therefore, even if one can black-box access a model for an unlimited number of times (full lookup table access; mathematically fully specified), having no access to the algorithmic implementation can make various computations prohibitive, rendering the model effectively unspecified under a computational budget. In this thesis, to "specify" a model not only means the mathematical specification but also algorithmic specification that may include the access to architecture, parameters, optimisation hyperparameters, training procedure, training data, and so on.

To represent the manipulator's knowledge on the specific algorithmic details of the target model, we review how the manipulation problem in an uncertain environment has been treated in general context. It is an important topic not only in our setup, but also in economics, business, physics, robotics, and many other quantitative disciplines where uncertainties arise naturally and should be embraced. In this section, we build connections between this vast literature and the data manipulation problem against learned models.

2.5.1 Representing uncertainty in other disciplines

The need for dealing with lack of knowledge arises in many disciplines – either due to the stochastic nature of measurements or due to adversarial effects. For optimisation theory, acknowledging the fact that certain optimisation parameters (e.g. cost vector in linear programming) may be not accurately specifiable, robust and stochastic optimisation (Ben-Tal *et al.*, 2009; Prékopa, 1995) have been proposed. Specifically, they treat parameters in optimisation as either a free parameter in a constrained space (robust optimisation) or a random variable (stochastic optimisation). Optimal control theory (Seierstad and Sydsaeter, 1986), which aims at obtaining an optimal sequence of decisions, also has the uncertainty-acknowledging version, robust control theory (Hansen and Sargent, 2001), which has gained popularity in econometrics. In certain situations, e.g. business competition, the uncertainty arises due to an "intelligent" activity of another agent. Game theory (Nash *et al.*, 1950) has been applied to these situations. Treatment of target uncertainty in Chapter 7 can be seen as an instance of game theory as well as robust optimisation.

2.5.2 Representing uncertainty for data manipulation

In the machine learning domain, several works have employed tools from the above theories to represent lack of knowledge on the *data manipulator*. Lanckriet *et al.* (2003) have proposed a robust optimisation type of training to represent uncertainty in the data generation process and have thus strengthened the model against inputs with both benign and adversarial perturbations. Brückner *et al.* (2012) have explicitly used game theoretical tools to strengthen a learned model against adversarial inputs. Nowadays, we see many machine learning security papers with direct reference to robust optimisation (Madry *et al.*, 2018; Kolter and Wong, 2017) or game theory (Dhillon *et al.*, 2018; Raval *et al.*, 2017). We believe game theory and robust optimisation are natural tools for analysing many data manipulation scenarios.

2.6 BLACK-BOX MODELS

One of the typical deployment scenarios for learned models is to limit the access to certain number of *black-box* queries: given input, returns output. This restricts the manipulator's knowledge on the algorithmic details of the network, including the architecture, parameters, optimisation procedure, and training data. As discussed earlier, this makes the exact gradient computation computationally prohibitive, making adversarial attacks less efficient.

In this section, we discuss prior work and our contribution on reverse-engineering the hidden algorithmic details from black-box neural networks. We then examine multiple prior works on attacking black-box neural networks with adversarial examples that give different efficiency-effectiveness trade-off, and discuss how the reverse-engineered hidden details can help generating more effective adversarial examples against the target.

2.6.1 Gaining knowledge on black-box models

Prior work on gaining knowledge on black-box models can be classified into two topics: *model extraction* and *membership inference*.

Model extraction. Model extraction methods either reconstruct the exact model parameters or build an *avatar model* that maximises the likelihood of the query input-output pairs from the target model (Tramer *et al.*, 2016; Papernot *et al.*, 2017). Tramer *et al.* (2016) have shown the efficacy of equation solving attacks and the avatar method in retrieving internal parameters of non-neural network models. Papernot *et al.* (2017) have also used the avatar approach with the end goal of generating adversarial examples. While the avatar approach first assumes model hyperparameters like a model family (architecture) and training data, our method in Chapter 8 (Oh *et al.*, 2018) predicts those hyperparameters themselves; our approach is complementary to the avatar approach.

Membership inference. Membership inference methods determine if a given data sample has been included in the training data (Ateniese *et al.*, 2015; Shokri *et al.*, 2017). Similarly to our work, Ateniese *et al.* (2015) have trained a decision tree metamodel over a set of classifiers trained on different datasets to determine the training dataset for a black-box model. Shokri *et al.* (2017) have developed notion of "shadow models" that recognise distinctive output patterns with respect to training versus non-training inputs. Chapter 8 considers not only inferring the training data dataset, but also exposing model architectures and optimisation procedures.

2.6.2 Attacking black-box models

Without the availability of cheap gradients, prior researches have proposed largely three paradigms for generating adversarial examples: numerical gradients, avatar model, and transferability.

Numerical gradients. Narodytska and Kasiviswanathan (2017) and Chen *et al.* (2017a) have proposed different modifications of numerical gradients to find adversarial perturbation directions for an input image. The caveat is that thousands and millions of queries are needed to compute a single adversarial example. The methods also do not scale well with the input dimensions.

Avatar model. These methods train a white box network that is supposedly similar to the target and generate perturbations against it (Papernot *et al.*, 2017, 2016a; Hayes and Danezis, 2017). Our black-box exposure method in Chapter 8 can complement this approach by e.g. determining the architecture for the avatar model.

Transferability. It has been shown that adversarial examples generated against one network can also fool other networks (Moosavi-Dezfooli *et al.*, 2017; Liu *et al.*, 2017a). Liu *et al.* (2017a) in particular have shown that generating adversarial examples against an ensemble of networks make it more transferable. We show in Chapter 8 (Oh *et al.*, 2018) that the adversarial examples transfer better within an architecture family (e.g. ResNet or DenseNet) than across, and that such a property can be exploited by our exposure technique to generate more powerful adversarial examples.

Part I

PRIVACY ANALYSIS IN VISUAL DATA

Compared to the significance and urgency, there are not many quantitative studies of user privacy in visual media. In this part, we focus on the identifiability of humans in personal photos in the context of state of the art recognition technologies.

In Chapter 3 (Oh *et al.*, 2015, 2017a) we develop a state of the art person recognition framework named `naeil`. `naeil` is able to reliably identify humans in the challenging personal photo setup where subjects appear in natural, uncooperative poses and appearance changes. Apart from achieving state of the art recognition performance, we contribute to the community by evaluating `naeil` under severe domain shifts (time and viewpoint). We show that `naeil` is robust with respect to such shifts. `naeil` will be used extensively throughout the thesis in privacy-relevant chapters as a target model to protect one's identity from.

In Chapter 4 (Oh *et al.*, 2016), we delve into common identity obfuscation techniques on photos: blurring or blacking-out faces, and removing identity tags. We demonstrate that `naeil`, our state of the art recognition system, is robust against such data manipulations and has the ability to adapt to the distributional shifts. We raise the public awareness of the consequences of sharing photos online and call for more effective obfuscation techniques.

3 PERSON RECOGNITION IN PERSONAL PHOTO COLLECTIONS

Before studying the visual privacy problem in depth, we first develop a person recognition system suitable for human identification in personal photos. Person recognition in social media photos sets new challenges for computer vision, including non-cooperative subjects (e.g. backward viewpoints, unusual poses) and great changes in appearance. To better understand the identifiability, we build a person recognition framework that leverages convnet features from multiple image regions (head, body, etc.). We verify that our simple approach achieves the state of the art result on the PIPA (Zhang *et al.*, 2015b) benchmark, arguably the largest social media based benchmark for person recognition to date with diverse poses, viewpoints, social groups, and events. We propose multiple recognition scenarios that enable the evaluation under the time and viewpoint based domain shifts. We present an in-depth analysis of the importance of different features according to time and viewpoint generalisability.

The chapter is based on Oh *et al.* (2015) and Oh *et al.* (2017a). As the first author, Seong Joon Oh has conducted all the experiments and was the main writer for the conference paper (Oh *et al.*, 2015) and the journal submission (Oh *et al.*, 2017a).

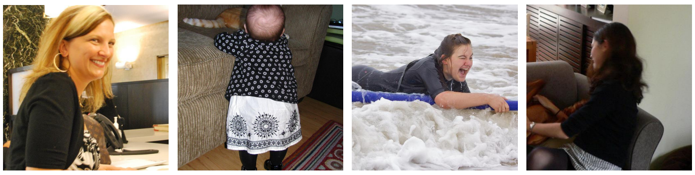
3.1 INTRODUCTION

With the advent of social media and the shift of image capturing mode from digital cameras to smartphones and life-logging devices, users share massive amounts of personal photos online these days. Automatic person identification in such photos is of great interest for social media users and companies hosting such services. Person recognition in personal photos is a relatively under-explored topic in computer vision with many new and interesting challenges: people may be focused on their activities with the face not visible, or can change clothing or hairstyle over time. See Figure 3.1 for the challenges of recognising a person in personal photo collections.

This chapter provides an in-depth study of human identifiability in challenging social media type of photos, as well as the simple yet effective person recognition systems `naeil` and `naeil2`. We combine ingredients from face recognition work (Huang *et al.*, 2007; Sun *et al.*, 2015) as well as body and scene cues based on simple convnet features.

The main contributions of the chapter are:

- Propose realistic and challenging person recognition scenarios on the PIPA benchmark (§3.3).



Head	✓	✗	✗	✗
Body	✓	✓	✗	✗
Attributes	✓	✓	✓	✗
More cues	✓	✓	✓	✓

Figure 3.1: In social media photos, face alone may not be an effective cue for recognition due to face occlusion and diverse poses. For example, the surfer (column 3) is only recognised when attribute cues are further considered.

- Provide a detailed analysis of the informativeness of different body regions and training data (§3.4).
- Verify that our final model `naeil2` achieves state of the art performance on PIPA (§3.5).
- Analyse the contribution of cues with respect to the time and viewpoint gap (§3.6).
- Discuss the performance of our methods under the open-world recognition setup (§3.7).

3.2 PIPA DATASET

Throughout the thesis, we will frequently refer to the PIPA dataset ("People In Photo Albums", (Zhang *et al.*, 2015b)) as the main testbed for experiments in person recognition. PIPA is, to the best of our knowledge, the largest dataset of social media type of photos with identity annotations (even for back view heads), capturing individuals in diverse social groups (e.g. friends, colleagues, family) and events (e.g. conference, vacation, wedding). The individuals also appear in diverse poses, point of view, activities, sceneries, and thus cover an interesting slice of the real world. Compared to previous social media datasets, such as Gallagher and Chen (2008) ($\sim$ 600 images, 32 identities), PIPA presents a leap both in size and diversity.

See Table 3 for the overview of PIPA statistics. PIPA features 37 107 Flickr personal photo album images (Creative Commons license), with 63 188 head bounding boxes of 2 356 identities. The heads are annotated with a bounding box and an identity tag. The head bounding boxes are tight around the skull, including the face and hair; occluded heads are hallucinated by the annotators. The dataset is partitioned into *train*, *val*, *test*, and *leftover* sets, with a rough ratio of 45 : 15 : 20 : 20 percent of the annotated heads. The leftover set is not used. Up to annotation errors, neither identities nor photo albums by the same uploader are shared among these sets.

	all	*train*	*val*	*test*	*leftover*
Photos	37 107	17 000	5 684	7 868	6 555
Albums	1 438	579	342	357	160
Instances	63 188	29 223	9 642	12 886	11 437
Identities	2 356	1 409	366	581	-

Table 3.1: PIPA dataset statistics (Zhang *et al.*, 2015c).

		val				*test*			
		$\mathcal{O}$	$\mathcal{A}$	$\mathcal{T}$	$\mathcal{D}$	$\mathcal{O}$	$\mathcal{A}$	$\mathcal{T}$	$\mathcal{D}$
spl.0	instance	4820	4859	4818	1076	6443	6497	6441	2484
	identity	366	366	366	65	581	581	581	199
spl.1	instance	4820	4783	4824	1076	6443	6389	6445	2485
	identity	366	366	366	65	581	581	581	199

Table 3.2: Split statistics for *val* and *test* sets. Total number of instances and identities for each split is shown.

3.2.1 Splits

We introduce different ways of splitting the training (gallery, *val*/*test*$_0$) and testing (probe, *val*/*test*$_1$) samples per identity. The split has direct impact on the difficulty of the person recognition problem. For example, it is harder to recognise a person who is wearing unusual clothing than to recognise someone in her "typical" clothing. Aiming to evaluate different levels of the generalisation ability, we introduce three new splits on top of the one provided in the PIPA dataset (see Original split below). Refer to Table 3.2 for the split statistics and Figure 3.2 for visualisations.

Original split $\mathcal{O}$. The Original split is proposed by Zhang *et al.* (2015b) and shares many similar examples per identity across the split – e.g. photos taken in a row. The Original split is thus easy - even nearest neighbour on raw RGB pixels works quite well (§3.5.1). In order to evaluate the ability to generalise across long-term appearance changes, we introduce three new splits below.

Album split $\mathcal{A}$. The Album split divides training and test samples for each identity according to the photo album metadata. Each split takes the albums while trying to match the number of samples per identity as well as the total number of samples across the splits. A few albums are shared between the splits in order to match the number of samples. Since the Flickr albums are user-defined and do not always strictly cluster events and occasions, the split may not be perfect.

Time split $\mathcal{T}$. The Time split divides the samples according to the time the photo was taken. For each identity, the samples are sorted according to their "photo-taken-date" metadata, and then divided according to the newest versus oldest basis. The

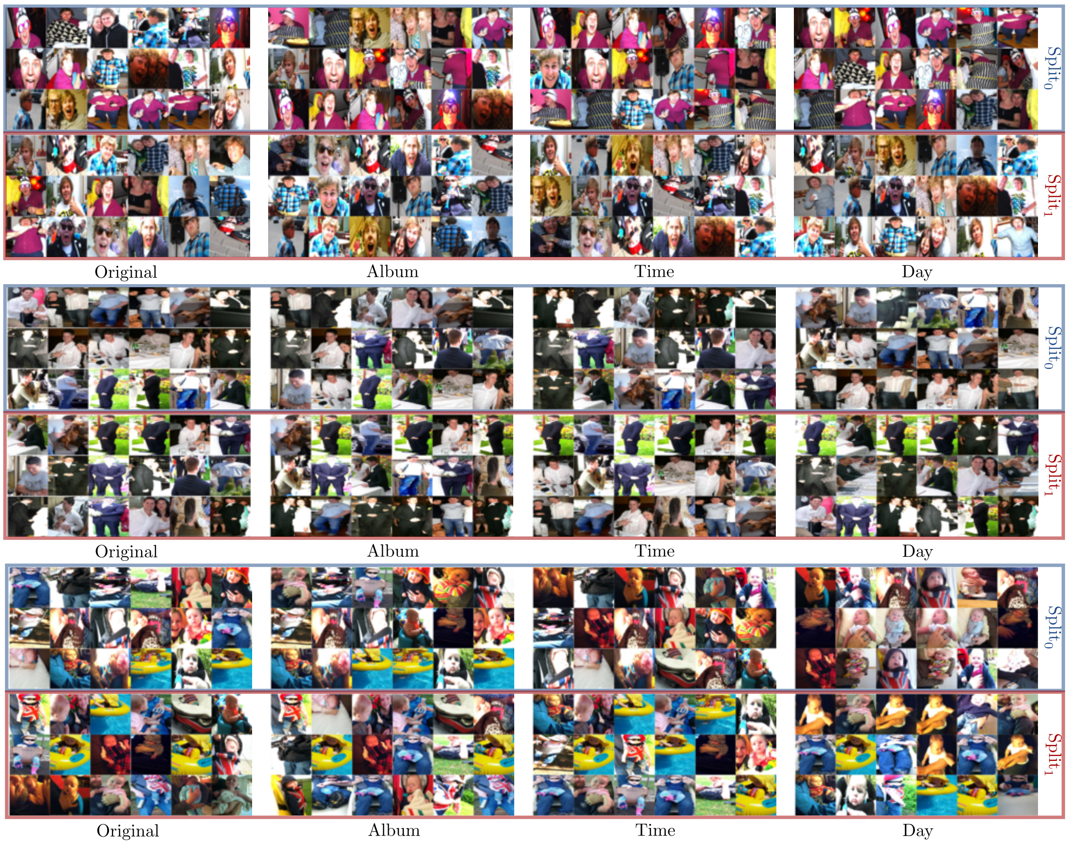


Figure 3.2: Visualisation of Original, Album, Time and Day splits for three identities (rows 1-3). A greater appearance gap is observed from Original to Day splits.

instances without time metadata are distributed evenly. This split evaluates the temporal generalisation of the recogniser. However, the "photo-taken-date" metadata is very noisy with lots of missing data.

Day split $\mathcal{D}$. The Day split divides the instances via visual inspection to ensure the firm "appearance change" across the splits. We define two criteria for division: (1) a firm evidence of date change such as {change of season, continent, event, co-occurring people} and/or (2) visible changes in {hairstyle, make-up, head or body wear}. We discard identities for whom such a division is not possible. After division, for each identity we randomly discard samples from the larger split until the sizes match. If the smaller split has $\leq$4 instances, we discard the identity altogether. The Day split enables clean experiments for evaluating the generalisation performance across strong appearance and event changes.

3.3 TASK AND EXPERIMENTAL SETUP

Task. At test time, the system is given a photo and ground truth head bounding box corresponding to the test instance (probe). The task is to choose the identity of the test instance among a given set of identities (gallery set, 200$\sim$500 identities) each with $\sim$10 training samples. In §3.7, we evaluate the methods when the test instance may be a background person (e.g. bystanders – no training image given). The system is then also required to determine if the given instance is among the seen identities (gallery set).

Protocol. We follow the PIPA protocol (Zhang *et al.*, 2015b) for data utilisation and model evaluation. The *train* set is used for convnet feature training. The *test* set contains the examples for the test identities. For each identity, the samples are divided into $test_0$ and $test_1$. For evaluation, we perform a two-fold cross validation by training on one of the splits and testing on the other. The *val* set is likewise split into val_0 and val_1, and is used for exploring different models and tuning hyperparameters.

Evaluation. We use the recognition rate (or accuracy), the rate of correct identity predictions among the test instances. For every experiment, we average two recognition rates obtained from the (gallery, probe) pairs (val_0, val_1) and (val_1, val_0) – analogously for *test*.

3.3.1 Face detection

Instances in PIPA are annotated by humans around their heads (tight around skull). We additionally compute face detections over PIPA to later compare the amount of identity information in head versus face (§3.4.6). On the other hand, we obtain the head orientation information as an auxiliary output of our face detections - this enables further analysis on the recognisability with respect to head orientations (§3.6). We also use the face detections to study the scenario without ground truth head box at test time (§3.7).

We use the open source DPM face detector (Mathias *et al.*, 2014). This detector is trained on $\sim$15k faces from the AFLW database, and is composed of 6 components which gives a rough indication of face orientation: $\pm 0°$ (frontal), $\pm 45°$ (diagonal left and right), and $\pm 90°$ (side views). Figure 3.4 shows example face detections on the PIPA dataset. It shows detections, the estimated orientation, the regressed head bounding box, the corresponding ground truth head box, and some failure modes. Faces corresponding to $\pm 0°$ are considered frontal (FR), and all others ($\pm 45°$, $\pm 90°$) are considered non-frontal (NFR). No ground truth is available to evaluate the face orientation estimation; except a few mistakes, the $\pm 0°$ components seems a rather reliable estimator (while more confusion is observed between $\pm 45° / \pm 90°$).

Given a set of detected faces (above certain detection score threshold) and the ground truth heads, the match is made according to the overlap (intersection over union) (Lin *et al.*, 2014). For matched heads, the corresponding face detections tell us which DPM component has fired, thereby allowing us to infer the head orientation (FR or NFR). Ground truth heads without face detections are referred to as "no face detected" or NFD. We denote detections without matching ground truth head as Background. See Figure 3.3 for visualisation.

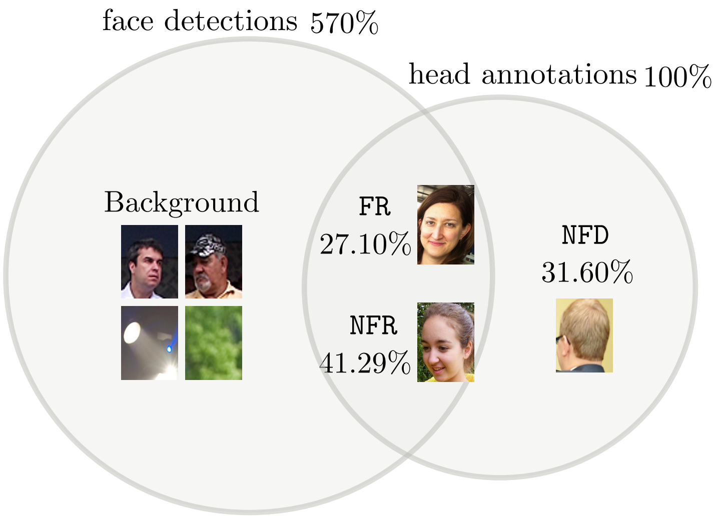
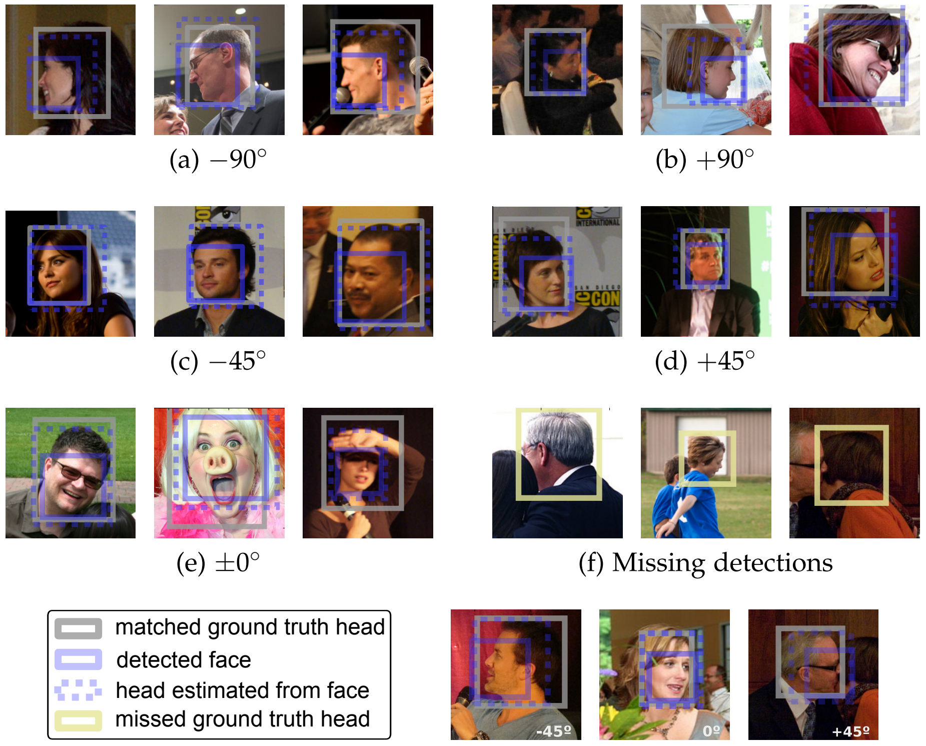


Figure 3.3: Face detections and head annotations in PIPA. The matches are determined by overlap (intersection over union). For matched faces (heads), the detector DPM component gives the orientation information (frontal versus non-frontal).



(a) −90°

(b) +90°

(c) −45°

(d) +45°

(e) ±0°

(f) Missing detections

(g) Legend

(h) Detected heads, but wrong orientation estimate

Figure 3.4: Example results from the face detector (PIPA *val* set), and estimated head boxes.

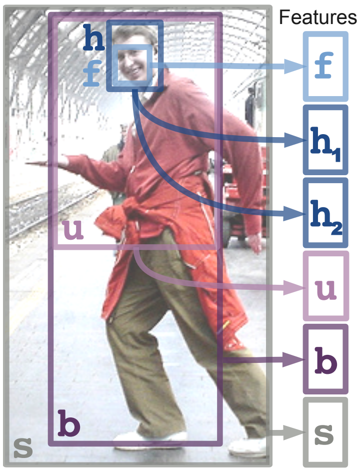


Figure 3.5: Regions considered for feature extraction: face f, head h, upper body u, full body b, and scene s. More than one cue can be extracted per region (e.g. $\mathtt{h}_1$, $\mathtt{h}_2$).

3.4 CUES FOR RECOGNITION

In this section, we investigate the cues for recognising people in social media photos. We begin with an overview of our model. Then, we experimentally answer the following questions: how informative are fixed body regions (no pose estimation) (§3.4.4)? How much does scene context help (§3.4.5)? Is it head or face (head minus hair and background) that is more informative (§3.4.6)? How much do we gain by using extended data (§3.4.7 & §3.4.8)? How effective is a specialised face recogniser (§3.4.10)? Studies in this section are based exclusively on the PIPA *val* set.

3.4.1 Model overview

At test time, given a ground truth head bounding box, we estimate five different regions depicted in Figure 3.5. Each region is fed into one or more convnets to obtain a set of cues. The cues are concatenated to form a feature vector describing the instance. Throughout the chapter we write + to denote vector concatenation. Linear SVM classifiers are trained over this feature vector (one versus the rest). In our final system, except for DeepID2+ (Sun *et al.*, 2015), all features are computed using the seventh layer (fc7) of AlexNet (Krizhevsky *et al.*, 2012) pre-trained for ImageNet classification. The cues only differ amongst each other on the image area and the fine-tuning used (type of data or surrogate task) to alter the AlexNet, except for the DeepID2+ feature.

3.4.2 Image regions used

We choose five different image regions based on the ground truth head annotation (given at test time, see the protocol in §3.3). The head rectangle h corresponds to the

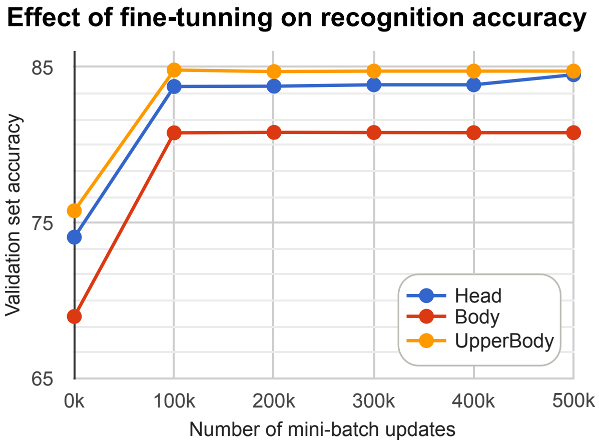


Figure 3.6: PIPA *val* set performance of different cues versus the SGD iterations in fine-tuning.

ground truth annotation. The full body rectangle b is defined as (3×head width, 6×head height), with the head at the top centre of the full body. The upper body rectangle u is the upper-half of b. The scene region s is the whole image containing the head.

The face region `f` is obtained using the DPM face detector discussed in §3.3.1. For head boxes with no matching detection (e.g. back views and occluded faces), we regress the face area from the head using the face-head displacement statistics on the PIPA *train* set. Five respective image regions are illustrated in Figure 3.5.

Note that the regions overlap with each other, and that depending on the person's pose they might be completely off. For example, b for a lying person is likely to contain more background than the actual body.

3.4.3 Fine-tuning and parameters

Unless specified otherwise AlexNet is fine-tuned using the PIPA *train* set (∼30k instances, ∼1.5k identities), cropped at five different image regions, with 300k mini-batch iterations (batch size 50). We refer to the base cue thus obtained as `f`, `h`, `u`, `b`, or `s`, depending on the cropped region. On the *val* set we found the fine-tuning to provide a systematic ∼10 percent points (pp) gain over the non-fine-tuned AlexNet (Figure 3.6). We use the seventh layer (fc7) of AlexNet for each cue (4 096 dimensions).

We train for each identity a one-versus-all SVM classifier with the regularisation parameter $C = 1$; it turned out to be an insensitive parameter in our preliminary experiments. As an alternative, the naive nearest neighbour classifier has also been considered. However, on the PIPA *val* set the SVMs consistently outperforms the NNs by a ∼10 pp margin.

	Cue	Accuracy
Chance level		1.0
Scene (§3.4.5)	s	27.1
Body	b	80.8
Upper body	u	84.8
Head	h	83.9
Face (§3.4.6)	f	74.5
Zoom out	f	74.5
	f+h	84.8
	f+h+u	90.7
	f+h+u+b	91.1
	f+h+u+b+s	91.2
Zoom in	s	27.1
	s+b	82.2
	s+b+u	86.4
	s+b+u+h	90.4
	s+b+u+h+f	91.2
Head+body	h+b	89.4
Full person	P = f+h+u+b	91.1
Full image	$\mathtt{P_s}$ = P+s	91.2

Table 3.3: PIPA *val* set accuracy of cues based on different image regions and their concatenations ("+" means concatenation).

3.4.4 How informative is each image region?

Table 3.3 shows the PIPA *val* set results of each region individually and in combination. Head h and upper body u are the strongest individual cues. Upper body is more reliable than the full body b because the lower body is commonly occluded or cut out of the frame, and thus is usually a distractor. Scene s is, unsurprisingly, the weakest individual cue, but it still useful information for person recognition (far above chance level). Importantly, we see that all cues complement each other, despite overlapping pixels. Overall, our features and combination strategy are effective.

3.4.5 Scene (s)

Other than a fine-tuned AlexNet we considered multiple feature types to encode the scene information. $\mathtt{s_{gist}}$: using the Gist descriptor (Oliva and Torralba, 2001) (512 dimensions). $\mathtt{s0_{places}}$: instead of using AlexNet pre-trained on ImageNet, we consider an AlexNet (PlacesNet) pre-trained on 205 scene categories of the "Places Database" (Zhou *et al.*, 2014) (~2.5 million images). $\mathtt{s_{places205}}$: Instead of the 4 096 dimensions PlacesNet feature, we also consider using the score vector for each scene category (205 dimensions). $\mathtt{s0}$,$\mathtt{s3}$: finally we consider using AlexNet in the same way as for body or head (with zero or 300k iterations of fine-tuning on the PIPA person recognition training set). $\mathtt{s3_{places}}$: $\mathtt{s0_{places}}$ fine-tuned for person recognition.

	Method	Accuracy
Gist	$\mathtt{s}_{\mathtt{gist}}$	21.6
PlacesNet scores	$\mathtt{s}_{\mathtt{places205}}$	21.4
raw PlacesNet	$\mathtt{s}_{0\mathtt{places}}$	27.4
PlacesNet fine-tuned	$\mathtt{s}_{3\mathtt{places}}$	25.6
raw AlexNet	$\mathtt{s}_0$	26.5
AlexNet fine-tuned	$\mathtt{s} = \mathtt{s}_3$	27.1

Table 3.4: PIPA *val* set accuracy of different scene cues. See descriptions in §3.4.5.

Results. Table 3.4 compares the different alternatives on the PIPA *val* set. The Gist descriptor $\mathtt{s}_{\mathtt{gist}}$ performs only slightly below the convnet options (we also tried the 4 608 dimensional version of Gist, obtaining worse results). Using the raw (and longer) feature vector of $\mathtt{s}_{0\mathtt{places}}$ is better than the class scores of $\mathtt{s}_{\mathtt{places205}}$. Interestingly, in this context pre-training for places classification is better than pre-training for objects classification ($\mathtt{s}_{0\mathtt{places}}$ versus $\mathtt{s}_0$). After fine-tuning $\mathtt{s}_3$ reaches a similar performance as $\mathtt{s}_{0\mathtt{places}}$.

Experiments trying different combinations indicate that there is little complementarity between these features. Since there is not a large difference between $\mathtt{s}_{0\mathtt{places}}$ and $\mathtt{s}_3$, for the sake of simplicity we use $\mathtt{s}_3$ as our scene cue $\mathtt{s}$ in all other experiments.

Conclusion. Scene $\mathtt{s}$ by itself, albeit weak, can obtain results far above the chance level. After fine-tuning, scene recognition as pre-training surrogate task (Zhou *et al.*, 2014) does not provide a clear gain over (ImageNet) object recognition.

3.4.6 Head ($\mathtt{h}$) or face ($\mathtt{f}$)?

A large portion of work on face recognition focuses on the face region specifically. In the context of photo albums, we aim to quantify how much information is available in the head versus the face region. As discussed in §3.3.1, we obtain the face regions $\mathtt{f}$ from a DPM face detector (Mathias *et al.*, 2014).

Results. There is a large gap of $\sim$10 percent points performance between $\mathtt{f}$ and $\mathtt{h}$ in Table 3.3 highlighting the importance of including the hair and background around the face.

Conclusion. Using $\mathtt{h}$ is more effective than $\mathtt{f}$, but $\mathtt{f}$ result still shows a fair performance. As with other body cues, there is a complementarity between $\mathtt{h}$ and $\mathtt{f}$; we suggest to use them together.

	Method	Accuracy
More data (§3.4.7)	$\texttt{h}$	83.9
	$\texttt{h} + \texttt{h}_{\texttt{cacd}}$	84.9
	$\texttt{h} + \texttt{h}_{\texttt{casia}}$	86.1
	$\texttt{h} + \texttt{h}_{\texttt{casia}} + \texttt{h}_{\texttt{cacd}}$	86.3
Attributes (§3.4.8)	$\texttt{h}_{\texttt{pipa11m}}$	74.6
	$\texttt{h}_{\texttt{pipa11}}$	81.7
	$\texttt{h} + \texttt{h}_{\texttt{pipa11}}$	85.0
	$\texttt{u}_{\texttt{peta5}}$	77.5
	$\texttt{u} + \texttt{u}_{\texttt{peta5}}$	85.2
	$\texttt{A} = \texttt{h}_{\texttt{pipa11}} + \texttt{u}_{\texttt{peta5}}$	86.2
	$\texttt{h} + \texttt{u}$	85.8
	$\texttt{h} + \texttt{u} + \texttt{A}$	90.1
naeil (§3.4.9)	naeil	91.7

Table 3.5: PIPA *val* set accuracy of different cues based on extended data.

3.4.7 Additional training data ($\texttt{h}_{\texttt{cacd}}, \texttt{h}_{\texttt{casia}}$)

It is well known that deep learning architectures benefit from additional data. DeepFace (Taigman *et al.*, 2014) used by PIPER (Zhang *et al.*, 2015b) is trained over $4.4 \cdot 10^6$ faces of $4 \cdot 10^3$ persons (the private SFC dataset, (Taigman *et al.*, 2014)). In comparison our cues are trained over ImageNet and PIPA's $29 \cdot 10^3$ faces over $1.4 \cdot 10^3$ persons. To measure the effect of training on larger data we consider fine-tuning using two open source face recognition datasets: CASIA-WebFace (CASIA, (Yi *et al.*, 2014b)) and the "Cross-Age Reference Coding Dataset" (CACD, (Chen *et al.*, 2014)).

CASIA contains $0.5 \cdot 10^6$ images of $10.5 \cdot 10^3$ persons (mainly actors and public figures). When fine-tuning AlexNet over these identities (using the head area $\texttt{h}$), we obtain the $\texttt{h}_{\texttt{casia}}$ cue.

CACD contains $160 \cdot 10^3$ faces of $2 \cdot 10^3$ persons with varying ages. Although smaller in total number of images than CASIA, CACD features greater number of samples per identity ($\sim 2\times$). The $\texttt{h}_{\texttt{cacd}}$ cue is built via the same procedure as $\texttt{h}_{\texttt{casia}}$.

Results. See the top part of Table 3.5 for the results. $\texttt{h} + \texttt{h}_{\texttt{cacd}}$ and $\texttt{h} + \texttt{h}_{\texttt{casia}}$ improve over $\texttt{h}$ (1.0 and 2.2 pp, respectively). Extra convnet training data seems to help. However, due to the mismatch in data distribution, $\texttt{h}_{\texttt{cacd}}$ and $\texttt{h}_{\texttt{casia}}$ on their own are about ~ 5 pp worse than $\texttt{h}$.

Conclusion. Extra convnet training data helps, even if they are from different types of photos.

3.4.8 Attributes ($\texttt{h}_{\texttt{pipa11}}$, $\texttt{u}_{\texttt{peta5}}$)

Albeit overall appearance might change day to day, one could expect that stable, long term attributes provide means for recognition. We build attribute cues by fine-tuning AlexNet features not for the person recognition task (like for all other cues), but rather for the attribute prediction surrogate task. We consider two sets attributes, one on the head region and the other on the upper body region.

We have annotated identities in the PIPA *train* and *val* sets ($1409 + 366$ in total) with five long term attributes: age, gender, glasses, hair colour, and hair length (see Table 3.6 for details). We build $\texttt{h}_{\texttt{pipa11}}$ by fine-tuning AlexNet features for the task of head attribute prediction.

For fine-tuning the attribute cue $\texttt{h}_{\texttt{pipa11}}$, we consider two approaches: training a single network for all attributes as a multi-label classification problem with the sigmoid cross entropy loss, or tuning one network per attribute separately and concatenating the feature vectors. The results on the PIPA *val* set indicate the latter ($\texttt{h}_{\texttt{pipa11}}$) performs better than the former ($\texttt{h}_{\texttt{pipa11m}}$).

For the upper body attribute features, we use the "PETA pedestrian attribute dataset" (Deng *et al.*, 2014). The dataset originally has 105 attributes annotations for $19 \cdot 10^3$ full-body pedestrian images. We chose the five long-term attributes for our study: gender, age (young adult, adult), black hair, and short hair (details in Table 3.6). We choose to use the upper-body $\texttt{u}$ rather than the full body $\texttt{b}$ for attribute prediction – the crops are less noisy. We train the AlexNet feature on upper body of PETA images with the attribute prediction task to obtain the cue $\texttt{u}_{\texttt{peta5}}$.

Results. See results in Table 3.5. Both PIPA ($\texttt{h}_{\texttt{pipa11}}$) and PETA ($\texttt{u}_{\texttt{peta5}}$) annotations behave similarly (~ 1 pp gain over $\texttt{h}$ and $\texttt{u}$), and show complementary (~ 5 pp gain over $\texttt{h}+\texttt{u}$). Amongst the attributes considered, gender contributes the most to improve recognition accuracy (for both attributes datasets).

Conclusion. Adding attribute information improves the performance.

3.4.9 Conference paper final model ($\texttt{naeil}$)

The final model in the conference paper (Oh *et al.*, 2015) combines five vanilla regional cues ($\texttt{P}_{\texttt{s}} = \texttt{P}+\texttt{s}$), two head cues trained with extra data ($\texttt{h}_{\texttt{cacd}}$,$\texttt{h}_{\texttt{casia}}$), and ten attribute cues ($\texttt{h}_{\texttt{pipa11}}$, $\texttt{u}_{\texttt{peta5}}$), resulting in 17 cues in total. We have named this method $\texttt{naeil}$[4].

Results. See Table 3.5 for the results. $\texttt{naeil}$, by combining all the cues considered naively, achieves the best result 91.7% on the PIPA *val* set.

[4] "naeil", 내일, means "tomorrow" and sounds like "nail".

Attribute	Classes	Criteria
Age	Infant	Not walking (due to young age)
	Child	Not fully grown body size
	Young Adult	Fully grown & Age $<$ 45
	Middle Age	45 $\leq$ Age $\leq$ 60
	Senior	Age$\geq$ 60
Gender	Female	Female looking
	Male	Male looking
Glasses	None	No eyewear
	Glasses	Transparant glasses
	Sunglasses	Glasses with eye occlusion
Haircolour	Black	Black
	White	Any hint of whiteness
	Others	Neither of the above
Hairlength	No hair	Absolutely no hair on the scalp
	Less hair	Hairless for $> \frac{1}{2}$ upper scalp
	Short hair	When straightened,$<$ 10 cm
	Med hair	When straightened, $<$chin level
	Long hair	When straightened, $>$chin level

Table 3.6: PIPA attributes details.

Conclusion. Cues considered thus far are complementary, and the combined model naeil is effective.

3.4.10 DeepID2+ face recognition module ($h_{\texttt{deepid}}$)

Face recognition performance has improved significantly in recent years with better architectures and larger open source datasets. In this section, we study how much face recognition helps in person recognition. While DeepFace (Taigman *et al.*, 2014) used by the PIPER (Zhang *et al.*, 2015b) would have enabled more direct comparison against PIPER, it is not publicly available. We thus choose the DeepID2+ face recogniser (Sun *et al.*, 2015). Face recognition technology is still improving quickly, and larger and larger face datasets are being released – the analysis in this section is thus an underestimate of current and future face recognisers.

The DeepID2+ network is a siamese neural network that takes 25 different crops of head as input, with the joint verification-identification loss. The training is based on large databases consisting of CelebFaces+ (Sun *et al.*, 2014), WDRef (Chen *et al.*, 2012), and LFW (Huang *et al.*, 2007) – totalling $2.9 \cdot 10^5$ faces of $1.2 \cdot 10^4$ persons. At test time, it ensembles the predictions from the 25 crop regions obtained by facial landmark detections. The resulting output is a 1 024 dimensional head feature that we denote as $h_{\texttt{deepid}}$.

Since the DeepID2+ pipeline begins with facial landmark detection, the DeepID2+ features are not available for instances for occluded or backward orientation heads. As a result, only 52 709 out of 63 188 instances (83.4%) have the DeepID2+ features

	Split			
Method	$\mathcal{O}$	$\mathcal{A}$	$\mathcal{T}$	$\mathcal{D}$
$\mathtt{h}$	83.9	77.9	70.4	40.7
$\mathtt{h}_{\mathtt{deepid}}$	68.5	66.9	64.2	60.5
$\mathtt{h} + \mathtt{h}_{\mathtt{deepid}}$	85.9	80.5	73.3	47.9
$\mathtt{h} \oplus \mathtt{h}_{\mathtt{deepid}}$	88.7	85.7	80.9	66.9
naeil	91.7	86.4	80.7	49.2
$\mathtt{naeil} + \mathtt{h}_{\mathtt{deepid}}$	92.1	86.8	81.1	51.0
naeil2	93.4	90.0	85.9	70.6

Table 3.7: PIPA *val* set accuracy of methods involving $\mathtt{h}_{\mathtt{deepid}}$. The optimal combination weights are $\lambda^\star = [0.60\ 1.05\ 1.00\ 1.50]$ for Original, Album, Time, and Day splits, respectively. "$\oplus$" means L_2 normalisation, and then concatenation.

available, and we use vectors of zeros as features for the rest.

Results - Original split. See Table 3.7 for the PIPA *val* set results for $\mathtt{h}_{\mathtt{deepid}}$ and related combinations. $\mathtt{h}_{\mathtt{deepid}}$ in itself is weak (68.5%) compared to the vanilla head feature h, due to the missing features for the back-views. However, when combined with h, the performance reaches 85.9% by exploiting information from strong DeepID2+ face features and the viewpoint robust h features.

Since the feature dimensions are not homogeneous (4 096 versus 1 024), we try L_2 normalisation of h and $\mathtt{h}_{\mathtt{deepid}}$ before concatenation ($\mathtt{h} \oplus \mathtt{h}_{\mathtt{deepid}}$). This gives a further 3 pp boost (88.7%) – better than $\mathtt{h} + \mathtt{h}_{\mathtt{cacd}} + \mathtt{h}_{\mathtt{casia}}$, the previous best model on the head region (86.3%).

Results - Album, Time and Day splits. Table 3.7 also shows results for the Album, Time, and Day splits on the PIPA *val* set. While the general head cue h degrades significantly on the Day split, $\mathtt{h}_{\mathtt{deepid}}$ is a reliable cue with roughly the same level of recognition in all four splits (60∼70%). This is not surprising, since face is largely invariant over time, compared to hair, clothing, and event.

On the other splits as well, the complementarity of h and $\mathtt{h}_{\mathtt{deepid}}$ is guaranteed only when they are L_2 normalised before concatenation. The L_2 normalised concatenation $\mathtt{h} \oplus \mathtt{h}_{\mathtt{deepid}}$ envelops the performance of individual cues on all splits.

Conclusion. DeepID2+, with face-specific architecture/loss and massive amount of training data, contributes highly useful information for the person recognition task. However, being only able to recognise face-visible instances, it needs to be combined with orientation-robust h to ensure the best performance. Unsurprisingly, having a specialised face recogniser helps more in the setup with larger appearance gap between training and testing samples (Album, Time, and Day splits). Better face recognisers will further improve the results in the future.

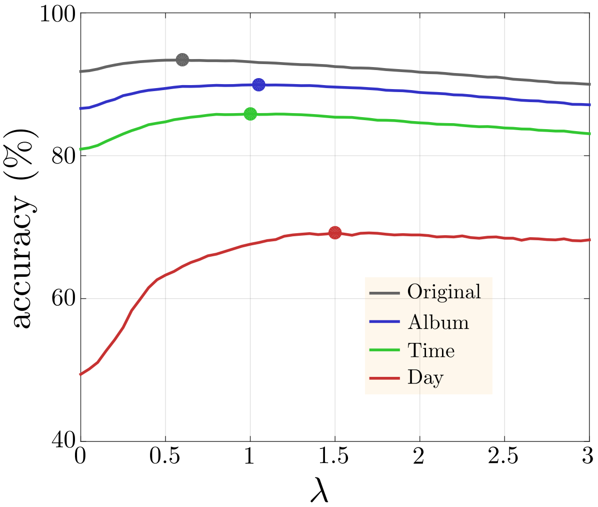


Figure 3.7: PIPA *val* set accuracy of $\texttt{naeil} \oplus_\lambda \texttt{h}_{\texttt{deepid}}$ for varying values of λ. Round dots denote the maximal *val* accuracy.

3.4.11 Journal submission final model (naeil2)

We build the final model of the journal submission, naeil2, by combining naeil and $\texttt{h}_{\texttt{deepid}}$. As seen in §3.4.10, naive concatenation is likely to fail due to even larger difference in dimensionality ($4\,096 \times 17 = 69\,632$ versus $1\,024$). We consider L_2 normalisation of naeil and $\texttt{h}_{\texttt{deepid}}$, and then performing a weighted concatenation.

$$\texttt{naeil} \oplus_\lambda \texttt{h}_{\texttt{deepid}} = \frac{\texttt{naeil}}{||\texttt{naeil}||_2} + \lambda \cdot \frac{\texttt{h}_{\texttt{deepid}}}{||\texttt{h}_{\texttt{deepid}}||_2}, \tag{3.1}$$

where, $\lambda > 0$ is a parameter and $+$ denotes a concatenation.

Optimisation of λ on PIPA *val* set. λ determines how much relative weight is to be given to $\texttt{h}_{\texttt{deepid}}$. As we have seen in §3.4.10, the amount of additional contribution from $\texttt{h}_{\texttt{deepid}}$ is different for each split. In this section, we find $\lambda^\star$, the optimal values for λ, for each split over the PIPA *val* set. The resulting combination of naeil and $\texttt{h}_{\texttt{deepid}}$ is our final method, naeil2. $\lambda^\star$ is searched on the equi-distanced points $\{0, 0.05, 0.1, \cdots, 3\}$.

See Figure 3.7 for the PIPA *val* set performance of $\texttt{naeil} \oplus_\lambda \texttt{h}_{\texttt{deepid}}$ with varying values of λ. The optimal weights are found at $\lambda^\star = [0.60\ 1.05\ 1.00\ 1.50]$ for Original, Album, Time, and Day splits, respectively. The relative importance of $\texttt{h}_{\texttt{deepid}}$ is greater on splits with larger appearance changes. For each split, we denote naeil2 as the combination naeil and $\texttt{h}_{\texttt{deepid}}$ based on the optimal weights.

Note that the performance curve is rather stable for $\lambda \geq 1.5$ in all splits. In practice, when the expected amount of appearance changes of subjects are unknown, our advice would be to choose $\lambda \approx 1.5$. Finally, we remark that the weighted sum can also be done for the 17 cues in naeil; finding the optimal cue weights is left as a future work.

Results. See Table 3.7 for the results of combining $\texttt{naeil}$ and $\texttt{h}_{\texttt{deepid}}$. Naively concatenated, $\texttt{naeil} + \texttt{h}_{\texttt{deepid}}$ performs worse than $\texttt{h}_{\texttt{deepid}}$ on the Day split (51.0% vs 60.5%). However, the weighted combination $\texttt{naeil2}$ achieves the best performance on all four splits.

Conclusion. When combining $\texttt{naeil}$ and $\texttt{h}_{\texttt{deepid}}$, a weighted combination is desirable, and the resulting final model $\texttt{naeil2}$ beats all the previously considered models on all four splits.

3.5 PIPA TEST SET RESULTS AND COMPARISON

In this section, we measure the performance of our final model and key intermediate results on the PIPA *test* set, and compare against the prior arts. See Table 3.8 for a summary.

3.5.1 Baselines

We consider two baselines for measuring the inherent difficulty of the task. The first baseline is the "chance level" classifier, which does not see the image content and simply picks the most commonly occurring class. It provides a lower bound for any recognition method, and gives a sense of how large the gallery set is.

Our second baseline is the raw RGB nearest neighbour classifier $\texttt{h}_{\texttt{rgb}}$. It uses the raw downsized (40×40 pixels) and blurred RGB head crop as feature. The identity of the Euclidean distance nearest neighbour training image is predicted at test time. By design, $\texttt{h}_{\texttt{rgb}}$ is only able to recognize near identical head crops across the $test_{0/1}$ splits.

Results. See results for "chance level" and $\texttt{h}_{\texttt{rgb}}$ in Table 3.8. While the "chance level" performance is low ($\leq$ 2% in all splits), we observe that $\texttt{h}_{\texttt{rgb}}$ performs unreasonably well on the Original split (33.8%). This shows that the Original split shares many nearly identical person instances across the split, and the task is very easy. On the harder splits, we see that the $\texttt{h}_{\texttt{rgb}}$ performance diminishes, reaching only 6.78% on the Day split. Recognition on the Day split is thus far less trivial – simply taking advantage of pixel value similarity would not work.

Conclusion. Although the gallery set is large enough, the task can be made arbitrarily easy by sharing many similar instances across the splits (Original split). We have remedied the issue by introducing three more challenging splits (Album, Time, and Day) on which the naive RGB baseline ($\texttt{h}_{\texttt{rgb}}$) no longer works (§3.2.1).

		Special modules		General features		Accuracy by split			
	Method	Face rec.	Pose est.	Data	Arch.	$\mathcal{O}$	$\mathcal{A}$	$\mathcal{T}$	$\mathcal{D}$
	Chance level	✗	✗	–	–	0.78	0.89	0.78	1.97
Head	$\mathtt{h}_{\mathtt{rgb}}$	✗	✗	–	–	33.8	27.2	16.9	6.8
	$\mathtt{h}$	✗	✗	I+P	Alex	76.4	67.5	57.1	36.5
	$\mathtt{h}+\mathtt{h}_{\mathtt{casia}}+\mathtt{h}_{\mathtt{cacd}}$	✗	✗	I+P+CC	Alex	80.3	72.8	63.2	45.5
	$\mathtt{h}_{\mathtt{deepid}}$	DeepID2+	✗	–	–	68.1	65.5	60.7	61.5
	$\mathtt{h} \oplus \mathtt{h}_{\mathtt{deepid}}$	DeepID2+	✗	I+P	Alex	85.9	82.0	75.9	66.0
	DeepFace	DeepFace	✗	–	–	46.7	–	–	–
Body	b	✗	✗	I+P	Alex	69.6	59.3	44.9	20.4
	h+b	✗	✗	I+P	Alex	83.4	74.0	63.0	38.2
	P = f+h+u+b	✗	✗	I+P	Alex	85.3	76.5	66.6	42.2
	GlobalModel	✗	✗	I+P	Alex	67.6	–	–	–
	PIPER	DeepFace	Poselets	I+P	Alex	83.1	–	–	–
	Pose	✗	Pose group	I+P+V	Alex	89.1	82.4	74.8	56.7
	COCO	✗	Part det.	I+P	Goog,Res	**92.8**	83.5	77.7	61.7
Image	$\mathtt{P}_{\mathtt{s}} = \mathtt{P}+\mathtt{s}$	✗	✗	I+P	Alex	85.7	76.7	66.6	42.3
	naeil (ours)	✗	✗	I+P+E	Alex	86.8	78.7	69.3	46.5
	Contextual	DeepID	✗	I+P	Alex	88.8	83.3	77.0	59.4
	naeil2 (ours)	DeepID2+	✗	I+P+E	Alex	90.4	**86.3**	**80.7**	**70.6**

	Terminology	Description
Method	GlobalModel	Zhang *et al.* (2015b)
	PIPER	Zhang *et al.* (2015b)
	Pose	Pose-aware person recognition (Kumar *et al.*, 2017)
	COCO	Congenerous cosine loss (Liu *et al.*, 2017b)
	naeil	Our conference paper method (Oh *et al.*, 2015)
	Contextual	Multi-level contextual model (Li *et al.*, 2016a)
	naeil2	Our journal submission method (Oh *et al.*, 2017a)
Modules	DeepID2+	Sun *et al.* (2015)
	DeepFace	Taigman *et al.* (2014)
	Poselets	Bourdev and Malik (2009)
	Part det.	Faster R-CNN (Shaoqing Ren, 2015)
	DeepID	Sun *et al.* (2014)
Data	I	ImageNet (Deng *et al.*, 2009) train set
	P	PIPA *train* set
	CC	CACD (Chen *et al.*, 2014) + CASIA (Yi *et al.*, 2014b)
	E	CC + PETA (Deng *et al.*, 2014)
	V	VGGFace (Parkhi *et al.*, 2015)
Arch.	Alex	AlexNet (Krizhevsky *et al.*, 2012)
	Goog	GoogleNetv3 (Szegedy *et al.*, 2016)
	Res	ResNet50 (He *et al.*, 2016)

Table 3.8: (Top) PIPA *test* set accuracy (%) of the proposed method and prior arts on the four splits. For each method, we indicate any face recognition or pose estimation module included, and the data and convnet architecture for other features. "$\oplus$" means concatenation after L_2 normalisation. (Bottom) Various terminologies in the table above are explained with references.

3.5.2 Methods based on head

We consider our four intermediate models (h, $\texttt{h}+\texttt{h}_{\texttt{casia}}+\texttt{h}_{\texttt{cacd}}$, $\texttt{h}_{\texttt{deepid}}$, $\texttt{h} \oplus \texttt{h}_{\texttt{deepid}}$) and the prior work DeepFace (Zhang *et al.*, 2015b; Taigman *et al.*, 2014).

We observe that, even without a specialised face module, h already performs better than DeepFace (76.4% versus 46.7%, Original split). We believe this is for two reasons: (1) DeepFace only takes face regions as input, leaving out valuable hair and background information (§3.4.6), (2) DeepFace only makes predictions on 52% of the instances where the face can be registered. Note that $\texttt{h}_{\texttt{deepid}}$ also does not always make predictions due to failures to estimate the pose (17% failure on PIPA), but performs better than DeepFace in the considered scenario (68.1% versus 46.7%, Original split).

3.5.3 Methods based on body

We consider three of our intermediate models (b, $\texttt{h}+\texttt{b}$, $\texttt{P} = \texttt{f}+\texttt{h}+\texttt{u}+\texttt{b}$) and four prior arts: GlobalModel (Zhang *et al.*, 2015b), PIPER (Zhang *et al.*, 2015b), Pose (Kumar *et al.*, 2017), COCO (Liu *et al.*, 2017b)). Pose and COCO appeared after the publication of the conference paper (Oh *et al.*, 2015). See Table 3.8 for the results.

Our body cue b and Zhang et al.'s GlobalModel are the same methods implemented independently. Unsurprisingly, they perform similarly (69.6% versus 67.6%, Original split).

Our $\texttt{h}+\texttt{b}$ method is the minimal system matching Zhang et al.'s PIPER (83.4% versus 83.1%, Original split). The feature vector of $\texttt{h}+\texttt{b}$ is about 50 times smaller than PIPER, and does not make use of a face recogniser or pose estimator.

In fact, PIPER captures the head region via one of its poselets. Thus, $\texttt{h}+\texttt{b}$ extracts cues from a subset of PIPER's "GlobalModel + Poselets" (Zhang *et al.*, 2015b), but performs better (83.4% versus 78.8%, Original split).

Methods since naeil. Pose by Kumar *et al.* (2017) uses extra keypoint annotations on the PIPA *train* set to generate pose clusters, and trains separate models for each pose cluster (PSM, pose-specific models). By performing a form of pose normalisation they have improved the results significantly: 2.3 pp and 10.2 pp over naeil on the Original and Day splits, respectively.

COCO (Liu *et al.*, 2017b) proposes a novel metric learning loss for the person recognition task. Metric learning gives an edge over classifier-based methods by enabling recognition of unseen identities without re-training. They further use Faster-RCNN detectors (Shaoqing Ren, 2015) to localise the face and body more accurately. The final performance is arguably good in all four splits, compared to Pose or naeil. However, one should note that the face, body, upper body, and full body features in COCO are based on GoogleNetv3 (Szegedy *et al.*, 2016) and ResNet50 (He *et al.*, 2016) – the numbers are not fully comparable to all the other methods that are largely based on AlexNet.

3.5.4 Methods based on full image

We consider our two intermediate models ($\mathtt{P_s} = \mathtt{P}+\mathtt{s}$, $\mathtt{naeil} = \mathtt{P_s}+\mathtt{E}$) and Contextual (Li *et al.*, 2016a), a method which appeared after the conference paper (Oh *et al.*, 2015).

Our naeil performs better than PIPER (Zhang *et al.*, 2015b) (86.8% versus 83.1%, Original split), while having a 6 times smaller feature vector and not relying on face recogniser or pose estimator.

Methods since naeil. Contextual by Li *et al.* (2016a) makes use of person co-occurrence statistics to improve the results. It performs 2.0 pp and 12.8 pp better than naeil on the Original and Day splits, respectively. However, one should note that Contextual employs the face recogniser DeepID (Sun *et al.*, 2014). We have found that a specialised face recogniser improves the recognition quality greatly on the Day split (§3.4.10).

3.5.5 Our final model naeil2

naeil2 is a weighted combination of naeil and $\mathtt{h}_{\mathtt{deepid}}$ (see §3.4.11 for details). Observe that, by attaching a face recogniser module on naeil, we achieve the best performance on Album, Time, and Day splits. In particular, on the Day split, naeil2 makes a 8.9 pp boost over the second best method COCO (Liu *et al.*, 2017b) (Table 3.8). On the Original split, COCO performs better (2.4 pp gap), but note that COCO uses more advanced feature representations (GoogleNet and ResNet).

Since naeil2 and COCO focus on orthogonal techniques, they can be combined to yield even better performances.

3.5.6 Computational cost

We report computational times for some pipelines in our method. The feature training takes 2-3 days on a single GPU machine. The SVM training takes 42 seconds for h (4 096 dim) and 1 118 seconds for naeil on the Original split (581 classes, 6 443 samples). Note that this corresponds to a realistic user scenario in a photo sharing service where $\sim$500 identities are known to the user and the average number of photos per identity is $\sim$10.

3.6 ANALYSIS

In this section, we provide a deeper analysis of individual cues towards the final performance. We measure how contributions from individual cues (e.g. face and scene) change when the system has to generalise across either time or head viewpoint. We study the performance as a function of the number of training samples per identity, and examine the distribution of identities according to their recognisability.

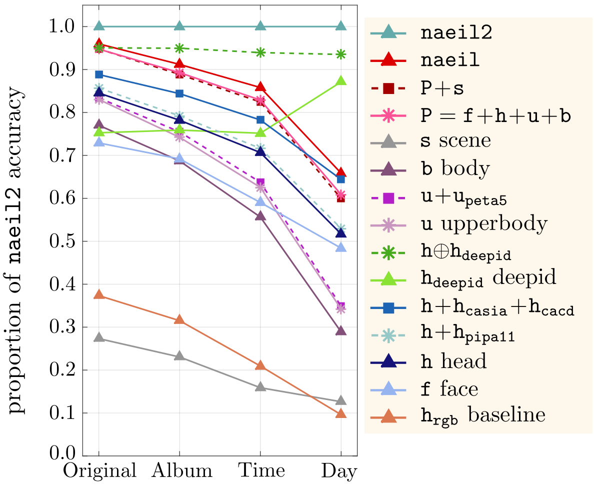


Figure 3.8: PIPA *test* set relative accuracy of various methods in the four splits, against the final system naeil2.

3.6.1 Contribution of individual cues under different time gaps

We measure the contribution of individual cues towards the final system naeil2 (§3.4.11) by dividing the accuracy for each intermediate method by the performance of naeil2. We report results in the four splits in order to determine which cues contribute more when there are larger time gap between training and testing samples and vice versa.

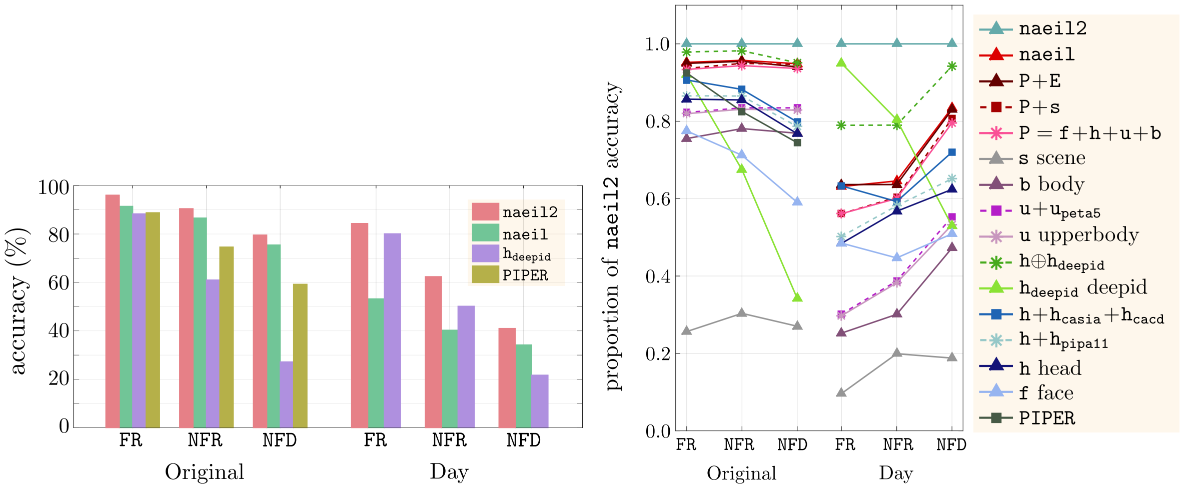
Results. See Figure 3.8 for the relative performances in four splits. The cues based more on context (e.g. b and s) see a greater drop from the Original to the Day split, whereas cues focused on face f and head h regions tend to drop less. Intuitively, this is due to the greater changes in clothing and events in the Day split.

On the other hand, h_{deepid} increases its relative contribution from Original to Day split, nearly explaining 90% of naeil2 in the Day split. h_{deepid} provides a valuable invariant face feature especially when the time gap is great. However, on the Original split h_{deepid} only reaches about 75% of naeil2. Head orientation robust naeil should be added to attain the best performance.

Conclusion. Cues involving context are stronger in the Original split; cues around face, especially the h_{deepid}, are robust in the Day split. Combining both types of cues yields the best performance over all considered time/appearance changes.

3.6.2 Performance by viewpoint

We study the impact of test instance viewpoint on the proposed systems. Cues relying on face are less likely to be robust to occluded faces, while body or context cues will be robust against viewpoint changes. We measure the performance of



(a) Accuracies.

(b) Accuracies relative to naeil2.

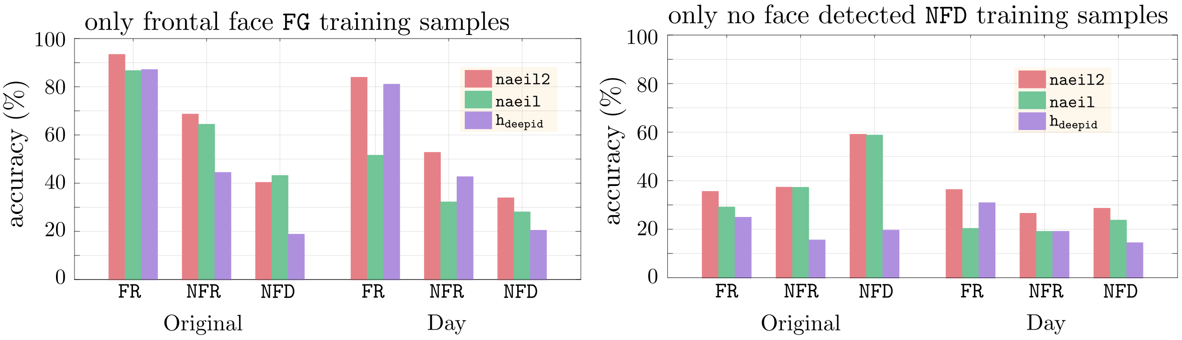
Figure 3.9: PIPA *test* set accuracy of methods on the frontal (FR), non-frontal (NFR), and no face detected (NFD) subsets.

models on the head orientation partitions defined by a DPM head detector (see §3.3.1): frontal FR, non-frontal NFR, and no face detected NFD. NFD subset is a proxy for back-view and occluded-face instances.

Results. Figure 3.9a shows the accuracy of methods on the three head orientation subsets for the Original and Day splits. All the considered methods show worse performance from frontal FR to non-frontal NFR and no face detected NFD subsets. However, in the Original split, naeil2 still robustly predicts the identities even for the NFD subset (∼80% accuracy). On the Day split, naeil2 also struggles on the NFD subset (∼20% accuracy). Recognition of NFD instances under the Day split constitutes the main remaining challenge of person recognition.

In order to measure contributions from individual cues in different head orientation subsets, we report the relative performance against the final model naeil2 in Figure 3.9b. The results are reported on the Original and Day splits. Generally, cues based on more context (e.g. b and .*s*) are more robust when the face is not visible than the face specific cues (e.g. f and h). Note that the h_{deepid} performance drops significantly in NDET, while naeil generally improves its relative performance in harder viewpoints. naeil2 envelops the performance of the individual cues in all orientation subsets.

Conclusion. naeil is more viewpoint robust than h_{deepid}, an opposite observation made for time-robustness analysis in §3.6.1. The combined model naeil2 takes the best of both worlds. The remaining challenge for person recognition lies on the no face detected NFD instances under the Day split. Perhaps image or social media metadata could be utilised (e.g. camera statistics, time and GPS location, social media friendship graph).



(a) Trained only on frontal (FR) (b) Trained only on no face detected (NFD)

Figure 3.10: PIPA *test* set performance when the identity classifier (SVM) is only trained on either frontal or no face detected subset. Related scenario: a robot has only seen frontal views of people; who is this person shown from the back view?

3.6.3 Generalisation across viewpoints

Here, we investigate the viewpoint generalisability of our models. For example, we challenge the system to identify a person from the back, having only shown frontal face samples.

Results. Figure 3.10 shows the accuracies of the methods, when they are trained either only on the frontal subset FR (left plot) or only on the no face detected subset NFD (right plot). When trained on FR, naeil2 has difficulties generalising to the NFD subset (FR versus NFD performance is ∼95% to ∼40% in Original; ∼85% to ∼35% in Day). However, the absolute performance is still far above the random chance (see §3.5.1), indicating that the learned identity representations are to a certain degree generalisable. The naeil features are more robust in this case than h_{deepid}, with a less dramatic drop from FR to NFD.

When no face is given during training (training on NFD subset), identities are much harder to learn in general. The recognition performance is low even for no-generalisation case: ∼60% and ∼30% for Original and Day, respectively, when trained and tested on NFD.

Conclusion. naeil2 does generalise marginally across viewpoints, largely attributing to the naeil features. It seems quite hard to learn identity specific features (either generalisable or not) from back-views or occluded faces (NFD).

3.6.4 Input resolution

This section provides analysis on the impact of input resolution. We aim to identify methods that are robust in different range of resolutions.

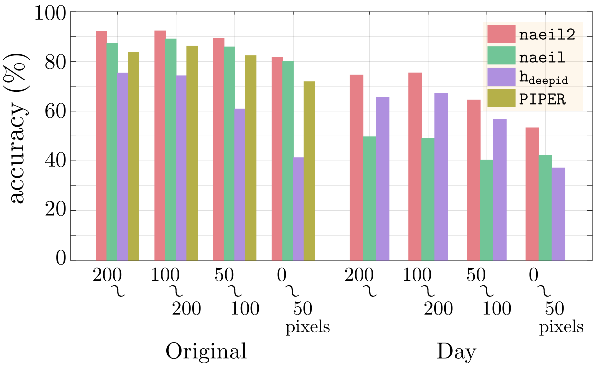


Figure 3.11: PIPA *test* set accuracy of systems at different levels of input resolution. Resolution is measured in terms of the head height (pixels).

Results. Figure 3.11 shows the performance with respect to the input resolution (head height in pixels). The final model naeil2 is robust against low input resolutions, reaching 80% even for instances with < 50 pixel heads on Original split. On the day split, naeil2 is less robust on low resolution examples (55%).

Component-wise, note that naeil performance is nearly invariant to the resolution level. naeil tends to be more robust for low resolution input than the h_{deepid} as it is based on body and context features and does not need high resolution faces.

Conclusion. For low resolution input naeil should be exploited, while for high resolution input h_{deepid} should be exploited. If unsure, naeil2 is a good choice – it envelops the performance of both in all resolution levels.

3.6.5 Number of training samples

We are interested in two questions: (1) if we had more samples per identity, would person recognition be solved with the current method? (2) how many examples per identity are enough to gather substantial amount of information about a person? To investigate the questions, we measure the performance of methods at different number of training samples per identity. We perform 10 independent sampling of the training samples at each level.

Results. Figure 3.12 shows the trend of recognition performances of methods with respect to different levels of training sample size. naeil2 saturates after $10 \sim 15$ training examples per person in Original and Day splits, reaching 92% and 83%, respectively, at 25 examples per identity. At the lower end, we observe that 1 example per identity is already enough to recognise a person far above the chance level (67% and 35% on Original and Day, respectively).

Conclusion. Adding a few times more examples per person will not push the performance to 100%. Methodological advances are required to fully solve the

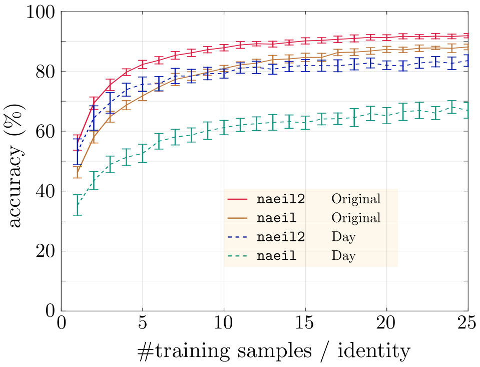
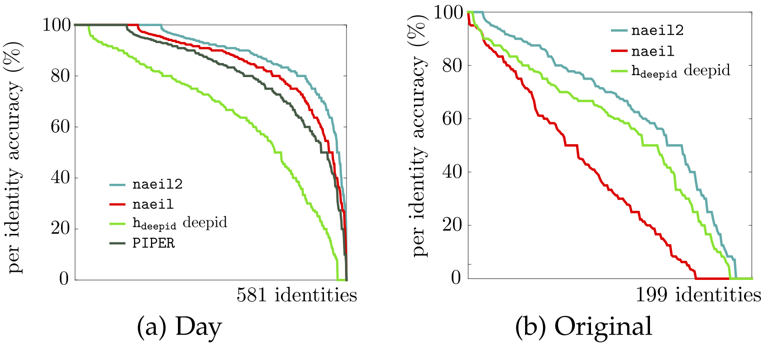


Figure 3.12: Accuracy versus # samples per identity (error bars: ±1 std dev).



(a) Day

(b) Original

Figure 3.13: Distribution of per-identity accuracies.

problem. On the other hand, the methods already collect a substantial amount of identity information only from a single sample per person (far above chance level).

3.6.6 Distribution of per-identity accuracy

Finally, we study which proportion of the identities are easy to recognise and how many are difficult. We study this by computing the distribution of identities according to their per-identity recognition accuracies.

Results. Figure 3.13 shows the per identity accuracy for each identity in a descending order for each considered method. On the Original split, naeil2 gives 100% accuracy for 185 out of the 581 test identities, whereas there was only one identity where the method totally fails. On the other hand, on the Day split there are 11 out of the 199 test identities for whom naeil2 achieves 100% accuracy and 12 identities with zero accuracy. In particular, naeil2 greatly improves the per-identity accuracy distribution over naeil, which gives zero prediction for 40 identities.

Conclusion. In the Original split, naeil2 is doing well on many of the identities already. In the Day split, the h_{deepid} feature has greatly improved the per-identity performances, but naeil2 still misses some identities. It is left as future work to focus on the hard identities.

3.7 OPEN-WORLD RECOGNITION

In the previous sections, we have assumed an oracle head bounding box at test time. We replace the head bounding box annotations with face detections in this section. In this scenario, probe images may contain identities that have not been seen in the gallery set - e.g. background people or even face-like objects. In this section, we introduce a simple thresholding trick on our person recognition system to enable it in this open-world scenario.

3.7.1 Method

At test time, body part crops are inferred from the detected face region (f). First, h is regressed from f, using the PIPA *train* set statistics on the scaling and displacement transformation from f to h. All the other regions (u, b, s) are computed based on h in the same way as in §3.4.2.

To measure if the inferred head region h is sound and compatible with the models trained on h (as well as u and b), we train the head model h on head annotations and test on the heads inferred from face detections. The recognition performance is 87.7%, while when trained and tested on the head annotations, the performance is 89.9%. We see a small drop, but not significant – the inferred regions to be largely compatible.

The gallery-background identity detection is done by thresholding the final SVM score output. Given a recognition system and test instance x, let $\mathcal{S}_k(x)$ be the SVM score for identity k. Then, we apply a thresholding parameter $\tau > 0$ to predict background if $\max_k \mathcal{S}_k(x) < \tau$, and predict the argmax gallery identity otherwise.

3.7.2 Evaluation metric

The evaluation metric should measure two aspects simultaneously: (1) ability to tell apart background identities and (2) ability to classify gallery identities. We first introduce a few terms to help defining the metrics. Refer to Figure 3.14 for a visualisation. We say a detected test instance x is a "foreground prediction" if $\max_k \mathcal{S}_k(x) \geq \tau$. A foreground prediction is either a true positive (TP) or a false positive (FP), depending on whether x is a gallery identity or not. If x is a TP, it is either a sound true positive TP_s or an unsound true positive TP_u, depending on the classification result $\arg\max_k \mathcal{S}_k(x)$. A false negative ($FN$) is incurred if a gallery identity is predicted as background.

We first measure the system's ability to screen background identities while at the same time classifying the gallery identities. The **recognition recall (RR)** at threshold τ is defined as follows

$$\text{RR}(\tau) = \frac{|TP_s|}{|\text{face det.} \cap \text{head anno.}|} = \frac{|TP_s|}{|TP \cup FN|}. \tag{3.2}$$

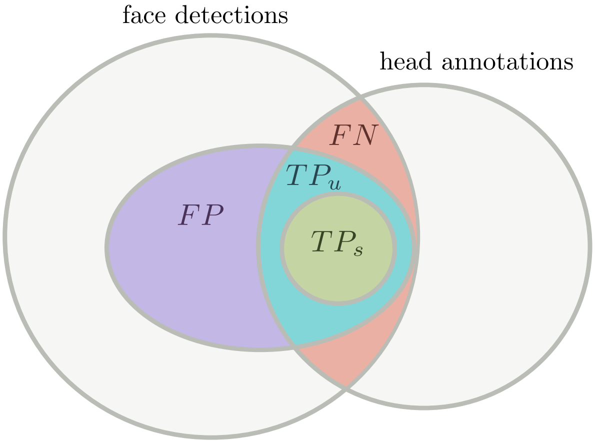


Figure 3.14: Diagram of various subsets generated by a person recognition system in an open world setting (cf. Figure 3.3). TP_s: sound true positive, TP_u: unsound true positive, FP: false positive, FN: false negative. See text for the definitions.

To factor out the performance of face detection, we constrain our evaluation to the intersection between face detections and head annotations (the denominator $TP \cup FN$). Note that the metric is a decreasing function of τ, and when $\tau \to -\infty$ the corresponding system is operating under the closed world assumption.

The system enjoys high RR when τ is decreased, but the system then predicts many background cases as foreground (FP). To quantify the trade-off we introduce a second metric: **false positive per image (FPPI)**. Given a threshold $\tau > 0$, FPPI is defined as

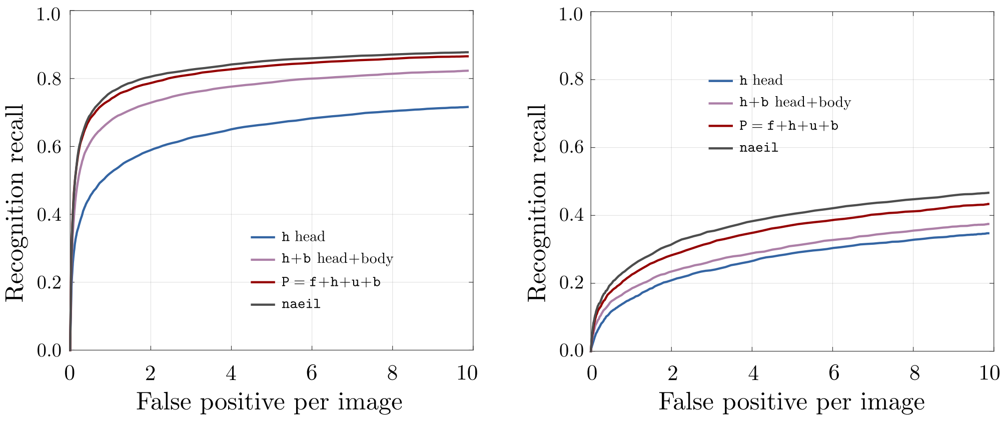
$$\text{FPPI}(\tau) = \frac{|FP|}{|\text{images}|}, \tag{3.3}$$

measuring how many wrong foreground predictions the system makes per image. It is also a decreasing function of τ. When $\tau \to \infty$, the FPPI attains zero.

3.7.3 Results

Figure 3.15 shows the recognition rate (RR) versus false positive per image (FPPI) curves parametrised by τ. As $\tau \to \infty$, $RR(\tau)$ approaches the close world performance on the face detected subset (FR ∪ NFR): 87.7% (Original) and 46.7% (Day) for naeil. In the open-world case, for example when the system makes one FPPI, the recognition recall for naeil is 76.3% (Original) and 25.3% (Day). Transitioning from the open world to close world, we see quite some drop, but one should note that the set of background face detections is more than 7× greater than the foreground faces.

Note that the DeepID2+ (Sun *et al.*, 2015) is not a public method, and so we cannot compute $\mathtt{h}_{\mathtt{deepid}}$ features ourselves; we have not included the $\mathtt{h}_{\mathtt{deepid}}$ or naeil2 results in this section.



(a) Original

(b) Day

Figure 3.15: Recognition recall (RR) versus false positive per image (FPPI) in open world.

3.7.4 Conclusion

A simple SVM score thresholding scheme can make our systems work in the open world recognition scenario with reasonable performances.

3.8 CONCLUSION

We have analysed the problem of person recognition in personal photo collections where people may appear with occluded faces, in diverse poses, and in various social events. We have investigated the efficacy of various cues, including the face recogniser DeepID2+ (Sun *et al.*, 2015), and their time and head viewpoint generalisability. For better analysis, we have contributed additional splits on PIPA (Zhang *et al.*, 2015b) that simulate different amounts of time gap between training and testing samples.

We make four major conclusions in this chapter. (1) Cues based on face and head are robust across time (§3.6.1). (2) Cues based on context are robust across head viewpoints (§3.6.2). (3) The final model naeil2, a combination of face and context cues, is robust across both time and viewpoint and achieves a ∼9 pp improvement over a recent state of the art approach on the challenging Day split (§3.5.5). (4) Better convnet architectures and face recognisers will improve the performance of the naeil and naeil2 frameworks in the future §3.5.5).

From the user privacy perspective, results in this chapter are alarming – current machine learning technology enables recognition of people from back-views, for example. In the next chapters, we will analyse the human identifiability further and propose novel obfuscation techniques.

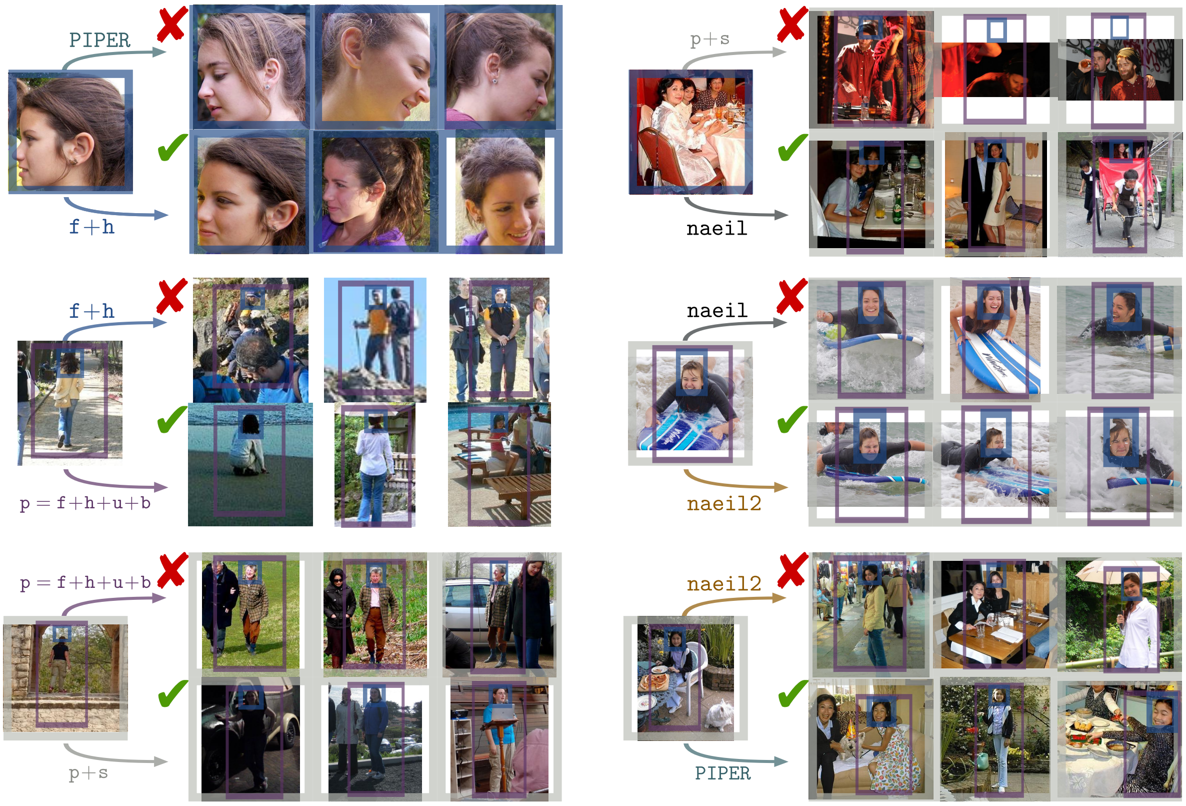


Figure 3.16: Various success and failure cases of intermediate and final systems (Original split). Given single probe images, we show gallery images of the predicted identities for corresponding recognition systems. Tick for correct, cross for wrong predictions.

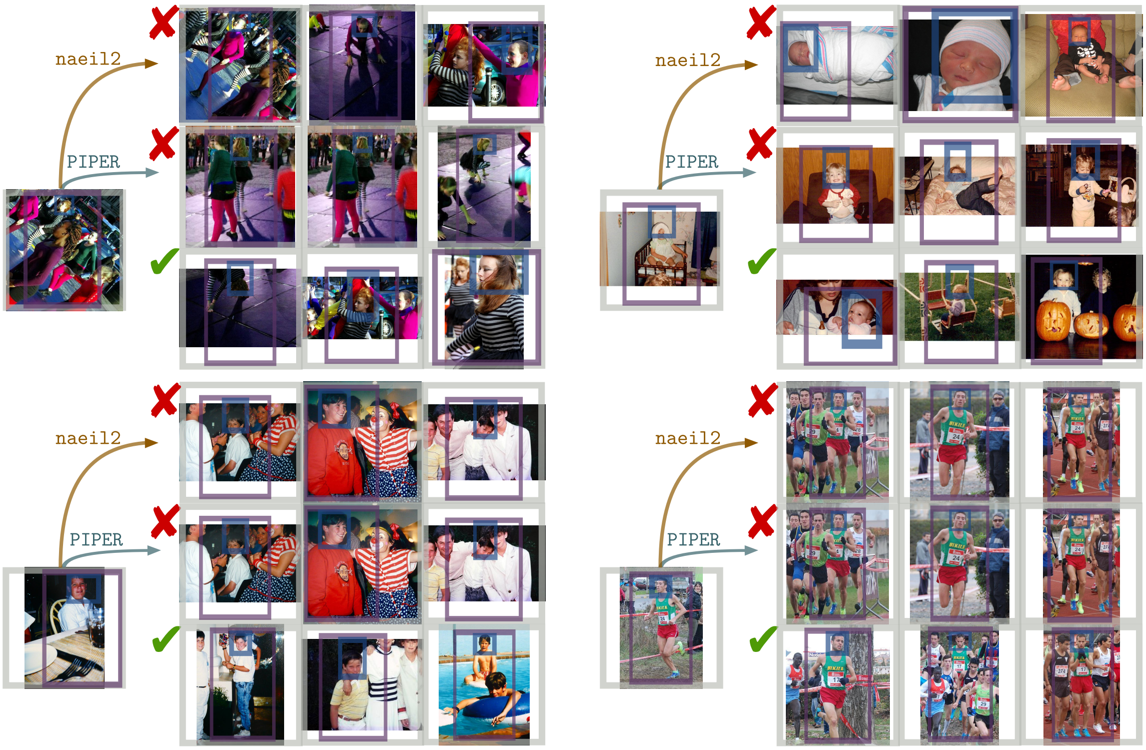


Figure 3.17: Failure cases of naeil2 and PIPER (Original split).

4 FACELESS PERSON RECOGNITION; PRIVACY IMPLICATIONS IN SOCIAL MEDIA

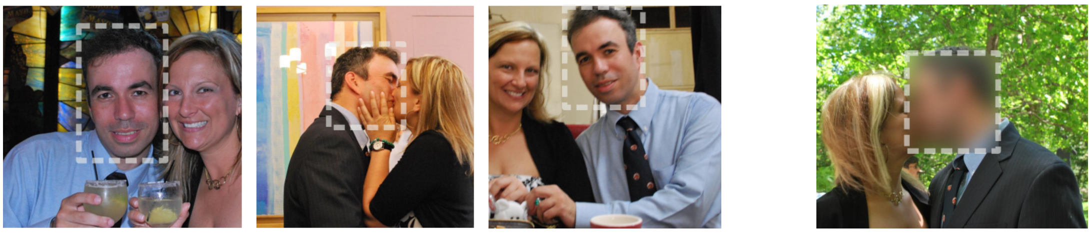
In the previous chapter, we have developed a person recogniser `naeil` and showed that it is robust against natural domain shifts, such and time and viewpoint changes, common in personal photo collections. In this section, we analyse the performance of `naeil` against *intentional* shifts in the distribution introduced by the users – e.g. face blurring or blacking out. In addition, we examine the effect of reducing the number of identity tags for each identity (another common obfuscation measure). We show that `naeil`, a state of the art system, is also robust against such commonplace image obfuscation and can moreover adapt to such distributional changes by re-training or by incorporating more contextual cues (e.g. from other photos). Results in this chapter further raise alertness in public as well as of the research community that more should be understood about the privacy implications of photo-sharing activities in the machine learning age.

The chapter is based on Oh *et al.* (2016). As the first author, Seong Joon Oh has conducted all the experiments and was the main writer of the manuscript.

4.1 INTRODUCTION

With the growth of the internet, more and more people share and disseminate large amounts of personal data be it on webpages, in social networks, or through personal communication. The steadily growing computation power, advances in machine learning, and the growth of the internet economy, have created strong revenue streams and a thriving industry built on monetising user data. It is clear that visual data contains private information, yet the privacy implications of this data dissemination are unclear, even for computer vision experts. We are aiming for a transparent and quantifiable understanding of the loss in privacy incurred by sharing personal data online, both for the uploader and other users who appear in the data.

In this work, we investigate the privacy implications of disseminating photos of people through social media. Although social media data allows to identify a person via different data types (timeline, geolocation, language, user profile, etc.) (Narayanan and Shmatikov, 2010), we focus on the pixel content of an image. We want to know how well a vision system can recognise a person in social photos (using the image content only), and how well users can control their privacy when limiting the number of tagged images or when adding varying degrees of obfuscation (see Figure 9.1) to their heads.



Person A training samples.

Is this person A ?

Figure 4.1: An illustration of one of the scenarios considered: can a vision system recognise that the person in the right image is the same as the tagged person in the left images, even when the head is obfuscated?

An important component to extract maximal information out of visual data in social networks is to fuse different data and provide a joint analysis. We propose our new Faceless Person Recogniser (described in §4.3), which not only reasons about individual images, but uses graph inference to deduce identities in a group of non-tagged images. We study the performance of our system on multiple privacy sensitive user scenarios (described in §4.2), and analyse the main results in §4.4. Since we focus on the image content itself, our results are a lower-bound on the privacy loss resulting from sharing such images. Our contributions are:

- We discuss dimensions that affect the privacy of users in online photos, and define a set of scenarios to study the question of privacy loss when social media images are aggregated and processed by a vision system.

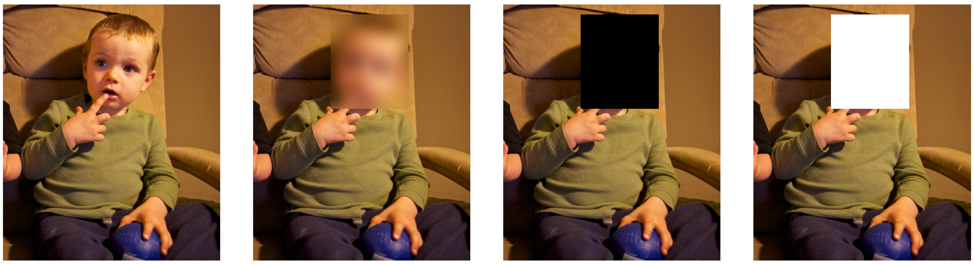
- We propose our new Faceless Person Recogniser that improves and builds on `naeil` (Chapter 3) to perform a joint inference across multiple photos.

- We measure the effectiveness of several common obfuscation scenarios against the Faceless Person Recogniser.

4.2 PRIVACY SCENARIOS AND SETUP

We consider a hypothetical social photo sharing service user. The user has a set of photos of herself and others in her account. Some of these photos have identity tags and others do not have such identity tags. We assume that all heads on the test photos have been detected, either by an automatic detection system, or because a user is querying the identity of a specific head. Note that we do not assume that the faces are visible nor that persons are in a frontal-upstanding pose. A "tag" is an association between a given head and a unique identifier linked to a specific identity (social media user profile).

Goal. The task of our recognition system is to identify a person of interest (marked via its head bounding box), by leveraging all the photos available (both with and



Fully visible Gaussian blur Black fill-in White fill-in

Figure 4.2: Obfuscation types considered.

without identity tags). In this work, we want to explore how effective different strategies are to protect the user identity.

4.2.1 Dimensions of recognisability

We consider four dimensions that affect how hard or easy it is to recognise a subject:

Number of tagged heads. We vary the number of tagged images available per identity. The more tagged images are available, the easier it should be to recognise someone in new photos. In our experiments, we assume that $1 \sim 10$ tagged images are available per person.

Obfuscation type. Users concerned with their privacy might take protective measures by blurring or masking their heads. Other than the fully visible case (non-obfuscated), we consider three other obfuscations types, shown in Figure 4.2. We consider both black and white, since Wilber *et al.* (2016) showed that commercial systems might react differently to these. The blurring parameters are chosen to resemble the YouTube face blur feature (Gaussian bandwidth $\sigma = \text{head height}/20 + 10$ pixels).

Amount of obfuscation. Depending on the user's activities (and her friends posting photos of her), not all photos might be obfuscated. We consider a variable fraction of these.

Domain shift. For the recognition task, there is a difference if all photos belong to the same event, where the appearance of people change little, or if the set of photos without tags correspond to a different event than the ones with identity tags. Recognising a person when the clothing, context, and illumination have changed ("across events") is more challenging than when they have not ("within events").

Abbre- viation	Brief description	Amount of tagged heads	Amount of obfuscated heads
S_0	Privacy indifferent	100%	0%
$\mathbf{S_1^{ø}}$	Some of my images are tagged	τ instances	0%
$\mathbf{S_2}$	One non-tagged head is obfuscated	10 instances	0%/1 instance
$\mathbf{S_3}$	All my heads are obfuscated	10 instances	100%
S_3'	All tagged heads are obfuscated	10 instances	100%/0%
S_3''	All non-tagged heads are obfuscated	10 instances	0%/100%

Table 4.1: Privacy scenarios considered. Each row in the table can be applied for the "across events" and "within events" case, and over different obfuscation types. See text §4.2.2. The obfuscation fraction indicates tagged/non-tagged heads. Bold abbreviations are reused in follow-up figures. In scenario S_1^τ, $\tau \in \{1.25, 2.5, 5, 10\}$.

4.2.2 Scenarios

Based on these four dimensions, we discuss a set of scenarios, summarised in Table 4.1. Clearly, these only cover a subset of all possible combinations along the mentioned four dimensions. However, we argue that this subset covers important and relevant aspects for our exploration on privacy implications.

Scenario $\mathbf{S_0}$. Here all heads are fully visible and tagged. Since all heads are tagged, the user is fully identifiable. This is the classic case without any privacy protection measure.

Scenario $\mathbf{S_1}$. There is no obfuscation but not all images are tagged. This is the scenario commonly considered for person recognition, e.g. (Gallagher and Chen, 2008; Zhang *et al.*, 2015b; Oh *et al.*, 2015). Unless otherwise specified we use S_1^{10}, where an average of 10 instances of the person are tagged (average across all identities). This is a common scenario for social media users, where some pictures are tagged, but many are not.

Scenario $\mathbf{S_2}$. Here the user has all of her heads visible, except for the one non-tagged head being queried. This would model the case where the user wants to conceal her identity in one particular published photo.

Scenario $\mathbf{S_3}$. The user aims at protecting her identity by obfuscating all her heads (using any obfuscation type, see Figure 4.2). Both tagged and non-tagged heads are obfuscated. This scenario models a privacy concerned user. Note that the body is still visible and thus usable to recognise the user.

Scenarios $\mathbf{S_3'}$&$\mathbf{S_3''}$. These consider the case of a user that inconsistently uses the obfuscation tactic to protect her identity. Albeit on the surface these seem like different scenarios, if the visual information of the heads cannot be propagated from/to the tagged/non-tagged heads, then these are functionally equivalent to **S3**.

Each of these scenarios can be applied for the "across/within events" dimension. In the following sections we will build a system able to recognise persons across these different scenarios, and quantify the effect of each dimension on the recognition capabilities (and thus their implication on privacy). For our system, the tagged heads become training data, while the non-tagged heads are used as test data.

4.2.3 Experimental setup

We investigate the scenarios proposed above through a set of controlled experiments on the PIPA dataset (§3.2, (Zhang *et al.*, 2015b)), which is by far the largest dataset of social media photos containing diverse social groups, events, and activities. We believe PIPA is the most realistic available testbed for privacy studies in social media photos. In this section, we project the discussed scenarios onto specific aspects of the PIPA dataset, describing how much realism can be achieved and what are possible limitations.

Data usage protocol. The PIPA dataset contains *train*, *val*, and *test* partitions, each containing disjoint sets of identities (§3.2). We use the *train* partition for convnet training and *val* for component-wise evaluation and hyperparameter search. The *test* partition is used for drawing final conclusions.

For each identity in the *val* and *test* partitions, we have further a partition of samples into splits $val/test_0$ and $val/test_1$. As done in Chapter 3, we regard one of the splits as the training samples (gallery) and the other as the test samples (probe). There are about 10 training and test samples for each identity on average. See §3.2 for further details on the splits.

In the context of the scenarios above, we consider $val/test_0$ as the set of tagged instances and $val/test_1$ as untagged ones.

Domain shift. §4.2.1 introduces the domain shift between tagged samples (gallery) and test samples (probe) as a factor for recognisability. To simulate the "within events" and "across events" scenarios, we use the Original ($\mathcal{O}$) and ($\mathcal{D}$) splits, respectively, as proxies. See §3.2.1 for more information on those splits.

Albums. Each photo in PIPA is associated with a Flickr album identifier. We use this photo album information during our graph inference (§4.3.3).

4.3 FACELESS RECOGNITION SYSTEM

In this section, we introduce the Faceless Recognition System (FRS) to study the effectiveness of privacy protective measures in §4.2. The system is built on top of the state of the art system `naeil` (Chapter 3). Unlike `naeil`, we enable reasoning across multiple photos. Human users do this naturally in a social media setup – when it is hard to recognise a person from a single photo due to e.g. face occlusion, users look for the person with a visible face in the other photos from the same album that are identified via e.g. cloth matching. Our system first builds a graph where each node represents an instance and the edges connect instances from the same albums. Then, FRS performs conditional random field (CRF) inference to jointly identify all the instances in the graph. Many previous works in computer vision have also used CRF inference for solving various joint prediction tasks (Gallagher and Chen, 2007; Stone *et al.*, 2008; Vu *et al.*, 2015; Hayder *et al.*, 2015). Our CRF inference is formulated as an optimisation problem:

$$\underset{Y}{\arg\max}\ \frac{1}{|V|}\sum_{i\in V}\phi_\theta(Y_i|X_i) + \frac{\alpha}{|E|}\sum_{(i,j)\in E} 1_{[Y_i=Y_j]}\psi_{\widetilde{\theta}}(X_i, X_j) \tag{4.1}$$

with observations X_i, identities Y_i and unary potentials $\phi_\theta(Y_i|X_i)$ defined on each node $i \in V$ (detailed in §4.3.1) as well as pairwise potentials $\psi_{\widetilde{\theta}}(X_i, X_j)$ defined on each edge $(i, j) \in E$ (detailed in §4.3.2). $1_{[\cdot]}$ is the indicator function, and $\alpha > 0$ controls the unary-pairwise balance.

We examine the unary ϕ_θ and pairwise $\psi_{\widetilde{\theta}}$ terms in greater detail in the following subsections.

4.3.1 Unary ϕ_θ: Single person recognition

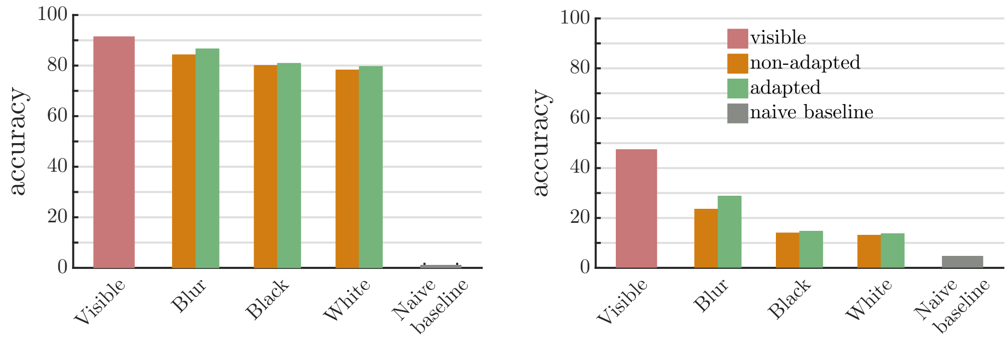
We build our unary ϕ_θ upon a state of the art, publicly available person recognition system, `naeil` (Chapter 3). The system was shown to be robust to decreasing numbers of tagged examples. Furthermore, as we will see, `naeil` achieves some robustness to face obfuscation techniques, as it not only uses the face but also context (e.g. body and scene) as cues.

For person instance X and identity Y, we define $\phi_\theta(Y|X)$ as the SVM score of X for identity Y. It comprises the unary term in Equation 4.1. When no pairwise term is present, Equation 4.1 boils down to $\underset{Y}{\arg\max}\ \phi_\theta(Y|X)$, the same person recognition scheme in Chapter 3.

We consider fine-tuning `naeil` with respect to obfuscation patterns (blacking or blurring heads), guiding `naeil` to focus more on e.g. body regions when the head cue is not reliable. For each obfuscation pattern, we train new `naeil` recognisers over obfuscated images (referred to as “adapted” in Figure 4.3) – both the feature extractor and identity prediction models are adapted. We assume that at test time these obfuscation patterns can easily be detected, and the corresponding model can be used.

Evaluation. We evaluate the performance of our single person recogniser under different obfuscation scenarios (head obfuscation patterns, domain gap, or varying number of training tags). See Figures 4.3 and 4.4 for the summary of our results.

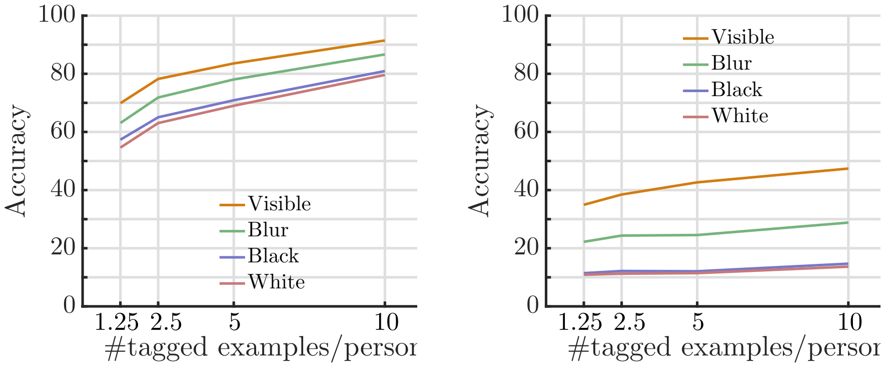
Figure 4.3 shows the effect of obfuscation (blur, black, or white) and the recogniser’s adaptation to those patterns (“adapted” versus “non-adapted”). We observe



(a) Within events

(b) Across events

Figure 4.3: Single person recogniser at different obfuscation types ("Adapted": models are fine-tuned for the corresponding obfuscation type).



(a) Within events

(b) Across events

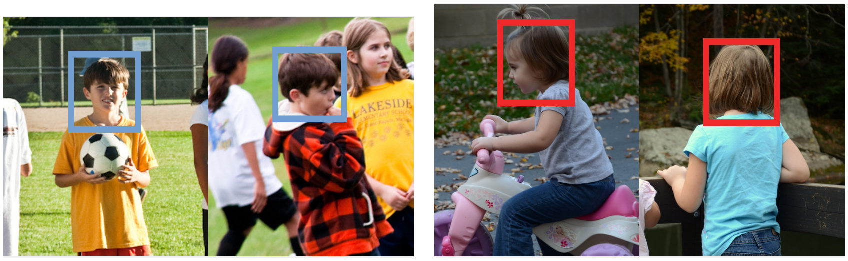
Figure 4.4: Single person recogniser at different tag rates.

that under "within events" case, the single person recogniser performance degrades only slightly by introducing obfuscation (from 91.5% to 84.3% for blur and 80.1% for black). The drop is greater for "across events" case: from 47.4% to 23.5% for blur and 14.0% for black. This is due to the fact that when events change, faces become the most reliable cue.

On the other hand, even after applying obfuscations, the recognition rate is still far above the chance level classifier. For "within events" performances on blur and black instances (84.3% and 80.1%, respectively) are still 80 times better than the chance-level accuracy of 1.0%. For the "across events" case, the recognition rate after black obfuscation is 14.0%, but this is still 3 times more accurate than the chance level of 4.7%.

We observe that adapting the recogniser to obfuscation patterns do improve its performance on the corresponding obfuscation patterns. The improvement is marginal for black or white obfuscation, but is substantial for blur, especially in the "across events" case (from 23.5% to 28.8%).

Wilber *et al.* (2016) have suggested that white obfuscation confuses a face detection system more than does the black. In our recognition setting, black and white fill-in



(a) Correct pair

(b) Incorrect pair

Figure 4.5: Person matching in social media photos is challenging.

have similar effects: 80.9% and 79.6% respectively ("within events", "adapted"). We omit the experiments for white fill-in obfuscation in the following sections.

We examine the effect of increasing or decreasing the number of identity tags on the single person recogniser performance in Figure 4.4. The system is surprisingly robust to the decreased number of tagged examples (gallery set) per identity. For example, in the "within events", non-obfuscated case, decreasing the number of tagged instances per identity from 10 to 1.25 only decreases the accuracy from 91.5% to 69.9%.

Conclusion. Blacking or blurring heads do not completely prevent recognition by a state of the art person recogniser, especially in the "within events" scenario. Moreover, by adapting the recogniser to obfuscation patterns, some of the performances can be re-gained. Black and white obfuscation types have similar effects against a person recogniser. Finally, the system is robust to greatly reducing the number of tags.

4.3.2 Pairwise $\psi_{\tilde{\theta}}$: Person pair matching

In this subsection, we describe and evaluate the pairwise term $\psi_{\tilde{\theta}}$ in the joint inference formulation in Equation 4.1. $\psi_{\tilde{\theta}}$ encodes the probability that the instances X_i and X_j contain the same identity. If $\psi_{\tilde{\theta}}(X_i, X_j)$ is high, then the CRF inference promotes joint outputs Y that predict the same identity for instances X_i and X_j. We expect that this formulation can strengthen weak unary predictions due to head obfuscation by propagating stronger predictions from the other photos with visible faces.

Person matcher. Given two instances X_i and X_j, we compute the probability that they contain the same identity. Note that the task of identity matching in social media photos is challenging due to heavy clothing changes and varying poses (see Figure 4.5).

The person matcher $\psi_{\tilde{\theta}}(X_i, X_j)$ is realised via a Siamese neural network on top of the feature vectors from X_i and X_j. In our case, we use only the head and body features, resulting in a 2×4096 dimensional feature for each instance. A Siamese network takes the inputs and processes each of them through three fully connected layers with ReLU activations with 2-dimensional output layer at the end. The outputs are then passed through a softmax function, leading to "match" and "non-match" probabilities. We indicate the "match" probability via $\psi_{\widetilde{\theta}}(X_i, X_j)$.

We train the person matcher on the PIPA *train* set, and then fine-tune it over the gallery instances of the test identities split_0. In order to simulate multiple scenarios, we train three types of person matchers: one for the visible pairs, one for the obfuscated pairs, and one for the mixed pairs. As for the unary term, we assume that the obfuscation patterns can be detected at test time and the corresponding pairwise model can be used.

Baseline unary based matcher. Instead of training a separate pairwise person matcher, one could use the unary scores to predict matches: if single person predictions for X_i and X_j coincide, predict "match", and vice versa. In order to produce an ROC curve, we introduce a continuous extension of the above procedure for match probability prediction.

We first compute the unary scores for the pair: $\phi_\theta(\cdot|X_i)$ and $\phi_\theta(\cdot|X_j)$. We then compute the mean unary entropy

$$\overline{H}\left(X_i, X_j\right) := \frac{1}{2}\big(H(\phi_\theta(\cdot|X_i)) + H(\phi_\theta(\cdot|X_j))\big) \tag{4.2}$$

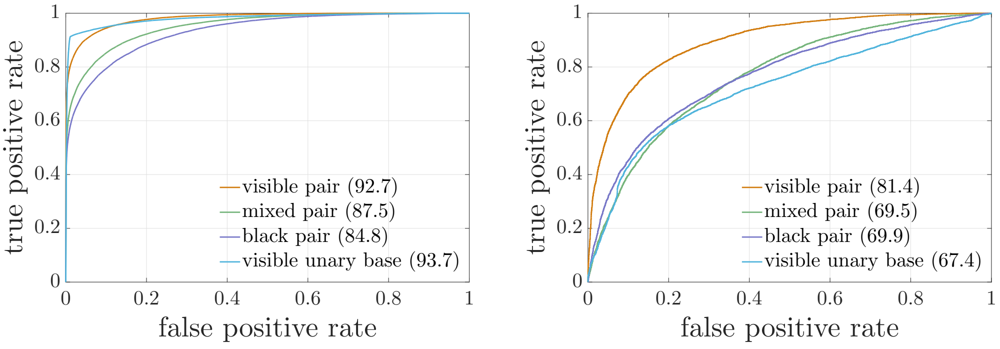
where the unary entropy $H(\phi_\theta(\cdot|X))$ is defined as $\sum_Y [-\phi_\theta(Y|X)\log(\phi_\theta(Y|X))]$. $\overline{H}\left(X_i, X_j\right)$ encodes the confidence of the joint unary predictions. Then, the match probability is computed via

$$\psi_{\text{unary}}\left(X_i, X_j\right) = \begin{cases} 1 - \frac{1}{2}\overline{H}\left(X_i, X_j\right) & \text{if unary predictions match} \\ \frac{1}{2}\overline{H}\left(X_i, X_j\right) & \text{otherwise} \end{cases} \tag{4.3}$$

In our preliminary evaluation, $\overline{H}\left(X_i, X_j\right)$ is typically less than 0.5. Thus, if the unary predictions match, then $\psi_{\text{unary}}\left(X_i, X_j\right)$ is within $[0.5, 1]$, with a lower value when the mean unary entropy (uncertainty) is higher. If the unary predictions do not match, then $\psi_{\text{unary}}\left(X_i, X_j\right)$ takes a value in $[0, 0.5]$ with a higher value for higher mean entropy (uncertainty). This provides a continuously relaxed version of the binary match based on unary argmax predictions.

Evaluation. We separately evaluate the matching performance of our person matcher $\psi_{\widetilde{\theta}}$ over the PIPA validation set probe image pairs (pairs in split_1). The performance is evaluated in the equal error rate (EER), the matching accuracy at the match score threshold where the false positive rate and the false negative rate meet.

See Figure 4.6 for the ROC curves for "within/across events" under three different obfuscation pair types (visible/mixed/black pairs). First of all, we notice that the matching performance for "within events" is far better than the "across events" case (92.7% versus 81.4% EER for visible pairs), and that the match accuracies drop as heads are obfuscated (albeit adapted against obfuscation patterns): from 92.7%



(a) Within events

(b) Across events

Figure 4.6: Person pair matching on the set of pairs in photo albums. The numbers in parentheses are the equal error rates (EER). The "visible unary base" refers to the baseline where the single person predictions are used to determine the match.

(visible) to 87.5% (mixed) and 84.8% (black) for "within events". However, even for the most severe scenario ("across events", mixed pair), the EER is 69.5%, far above the chance level of 50%. The person matcher performance seems reasonable.

We note that fine-tuning the matcher over the test identities split_0 is crucial. For the visible pairs, EER improves from 79.1% to 92.7% in the "within events", and from 74.5% to 81.4% in "across events".

Finally, the unary baseline performs marginally better than the visible pair model under the "within events": 93.7% versus 92.7%. Under the "across events", on the other hand, the visible pair model beats the baseline by a large margin: 81.4% versus 67.4% (Figure 4.6). In practice, the system has no information whether the query image is from within or across events. We thus use the Siamese person matcher, which beats the unary baseline on average.

Conclusion. Our person matcher achieves a reasonable performance in matching identities in a challenging social media photos, and beats the unary-based baseline. Fine-tuning the matcher over some examples from the test identities (split_0) is crucial.

4.3.3 Graph inference

Given the unary term from §4.3.1 and pairwise term from §4.3.2, we perform the CRF inference in Equation 4.1 to predict identities for a set of probe images. This subsection provides a detailed description of the inference procedure and intermediate empirical analyses.

Graph building. We describe how the node test instances (probe images or split_1) are interconnected in our inference graph. For both efficiency and performance, we do not build a complete graph over all the test instances; we only connect within albums, information given in the form of Flickr metadata for PIPA images (Zhang

Setting	Test/Val	#classes	#nodes	#edges	#albums
Within events	Test	581	6 443	252 431	351
	Val	366	4 820	228 116	300
Across events	Test	199	2 485	51 633	192
	Val	65	1 076	17 095	137

Table 4.2: Problem size for the CRF inference.

Setting (scenario=$\mathbf{S_1}$)	Inference	Time	Accuracy
Within events	Unary only	-	91.5
	Tree approximation	714 sec	91.8
	Max-product	15 hrs	91.4
Across events	Unary only	-	47.4
	Tree approximation	5 sec	55.0
	Max-product	87 sec	52.2

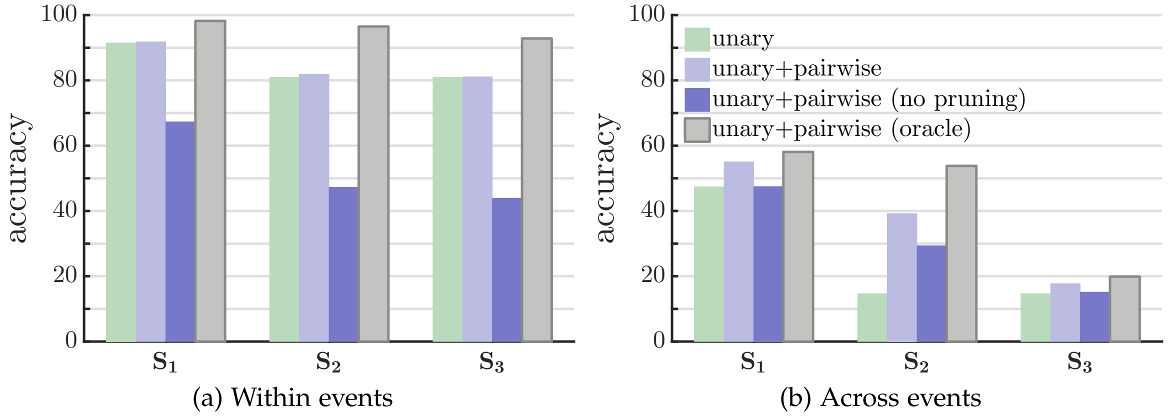
Table 4.3: Computational time and accuracy for inference algorithms.

et al., 2015b). We remark that many social media platforms employ analogues of the "album" in Flickr, user-specified groups of photos with similar locations, events, and contexts. We present a detailed report on the graph size of our particular experiment in Table 4.2.

Edge pruning. We consider a heuristic for pruning edges of the graph based on the pairwise scores. Since the person matcher is not perfect (Figure 4.6), it could benefit to prune edges with low confidence predictions. Through our preliminary evaluation, we have found that pruning negatively matched pairs from the inference results in a better performance. We will compare the performance with and without the negative edge pruning later.

Inference. The CRF inference involves one hyperparameter α determining the relative weights for the unary and pairwise terms. A preliminary experiment on the impact of α on the validation set person recognition performance shows that for both "within" and "across events", the validation performance plateaus for $\alpha \geq 100$. We use $\alpha = 100$ for all the experiments.

For efficiency, we consider performing an approximate inference. Given a node to infer identity, we consider propagations only on the neighbouring edges for the node. Since the resulting graph is a tree, this significantly reduces the computation time, while achieving similar or even better accuracy than the full max-product inference. See Table 4.3 for a preliminary results on the computational time and performance with and without this approximate inference. For "within events", the reduction in inference time for the whole validation set is from 15 hours to only 714 seconds. The graph inference is implemented via PyStruct (Müller and Behnke, 2014).



(a) Within events

(b) Across events

Figure 4.7: Validation performance of the CRF joint inference in three scenarios, **S1**, **S2**, and **S3** (see §4.2), under black fill-in obfuscation.

Oracle pairwise. To factor out the errors stemming from the wrong pairwise predictions, we build an oracle case that assumes perfect pairwise potentials ($\psi_{\widetilde{\theta}}(X_i, X_j) = 1_{\left[Y_i = Y_j\right]}$, where $1_{[\cdot]}$ is the indicator function and Y are the ground truth identities). We do not perform negative edge pruning here.

Evaluation. The results of the joint inference (for the black obfuscation case) are presented in Figure 4.7. Shown results include unary-only case, final system without the negative edge pruning, final system ("unary+pairwise"), and the final system with oracle pairwise terms. Performances are measured under "within/across events" and with respect to multiple obfuscation setups **S1**, **S2**, and **S3** (see §4.2).

After graph inference, all scenarios in the "within event" case reach recognition rates above 80%. For "across events", both S_1 and S_2 are above 35%. Compared to the unary-only case ("unary"), "unary+pairwise" performs better in general, with quite strong boost for "across events", **S2** case (one obfuscated test instance): accuracy from 15% to 39%. In this case, since only one test head is obfuscated, good recognition results from other instances get propagated to the obfuscated instance through the graph inference.

We observe that the negative edge pruning turns out to be quite important. Without pruning, the pairwise reasoning can even harm the recognition accuracy. For "within events" **S1**, the accuracy drops from 91.5% to 67.3% by adding both positive and negative edges. This is due to the erroneous match predictions. This is confirmed again by the oracle pairwise experiments, in which case we do not prune any edge from the graph. When pairwise terms are perfect, the joint inference results in the best performance in all setups considered, enveloping the unary-only and the negative edge pruned joint inference performances. The need for negative edge pruning is really due to the imperfect match predictions, and as more effective person matchers are developed, the overall performance will improve even without the pruning.

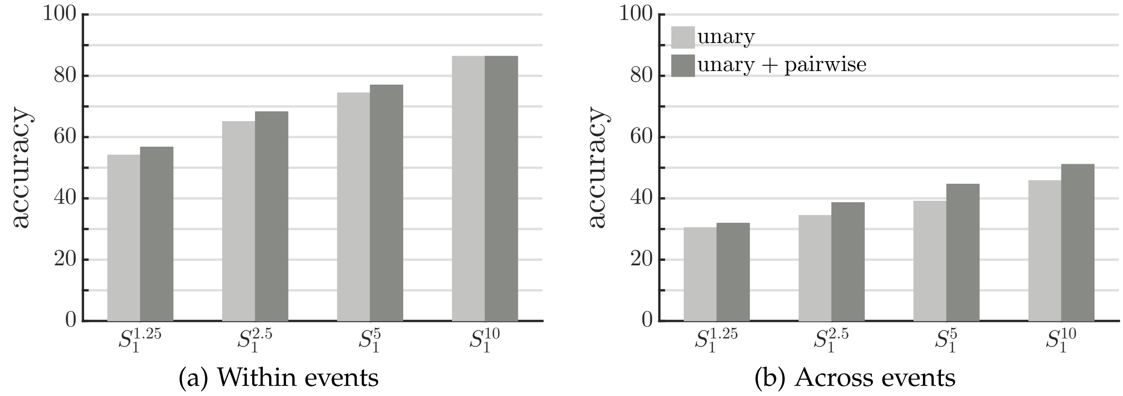
Conclusion. Joint inference performance supersedes single-instance based performances, if negative edges are properly pruned to take care of erroneous pairwise predictions. Oracle pairwise term experiments indicate that there is more potential for performance gain once the person matchers are improved.

4.4 TEST SET RESULTS & ANALYSIS

Following the experimental protocol in §4.2.3, we now evaluate our Faceless Recognition System on the PIPA *test* set. The main results are summarised in Figures 4.8 and 4.9. Figure 4.10 shows some qualitative results over the test set.

Amount of tagged heads. Figure 4.8 shows that even with only 1.25 tagged photos per person on average, the system can recognise users far better than chance level (naive baseline; best guess before looking at the image). Even with such little amount of training data, the system predicts 56.8% of the instances correctly within events and 31.9% across events; which is 73× and 16× higher than chance level, respectively. We see that even few tags provide a threat for privacy and thus users concerned with their privacy should avoid having (any of) their photos tagged.

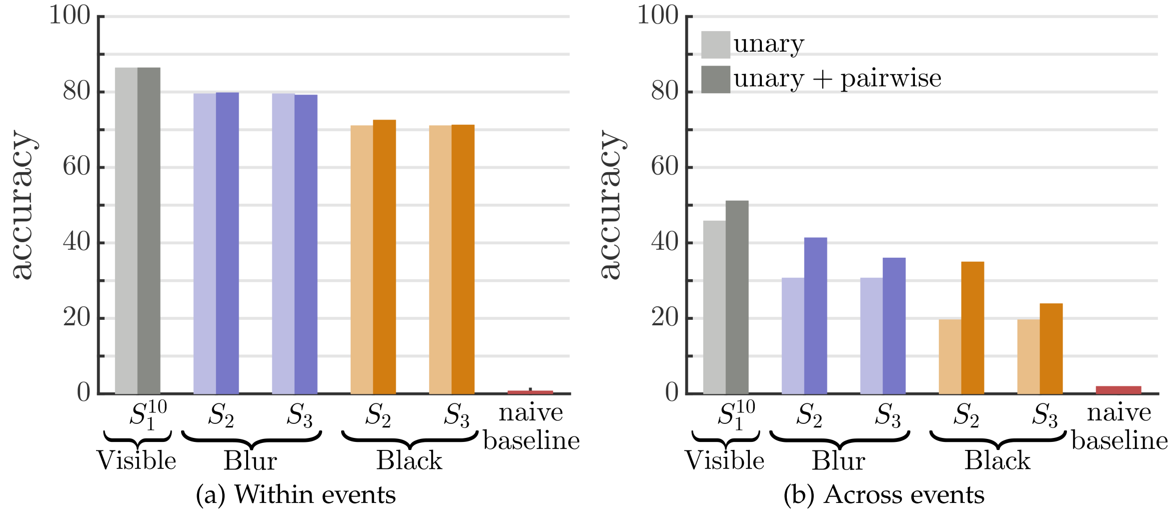
Obfuscation type. For both scenario $\mathbf{S_2}$ and $\mathbf{S_3}$, Figure 4.9 (and the results from §4.3.1) indicates the same privacy protection ranking for the different obfuscation types. From higher protection to lower protection, we have Black ≈ White > Blur > Visible. Albeit blurring does provide some protection, the machine learning algorithm still extracts useful information from that region. When our full Faceless Recognition System is in use, one can see that (Figure 4.9) obfuscation helps, but only to a limited degree: e.g. 86.4% ($\mathbf{S_1}$) to 71.3% ($\mathbf{S_3}$) under within events and 51.1% ($\mathbf{S_1}$) to 23.9% ($\mathbf{S_3}$) under across events.



(a) Within events

(b) Across events

Figure 4.8: Impact of number of tagged examples: $\mathbf{S_1^{1.25}}$, $\mathbf{S_1^{2.5}}$, $\mathbf{S_1^{5}}$, and $\mathbf{S_1^{10}}$.



(a) Within events

(b) Across events

Figure 4.9: Co-recognition results for scenarios $\mathbf{S_1^{10}}$, $\mathbf{S_2}$, and $\mathbf{S_3}$ with black fill-in and Gaussian blur obfuscations (white fill-in match black results).

Amount of obfuscation. We cover three scenarios: every head fully visible ($\mathbf{S_1}$), only the test head obfuscated ($\mathbf{S_2}$), and every head fully obfuscated ($\mathbf{S_3}$). Figure 4.9 shows that within events obfuscating either one ($\mathbf{S_2}$) or all ($\mathbf{S_3}$) heads is not very effective, compared to the across events case, where one can see larger drops for $\mathbf{S_1} \to \mathbf{S_2}$ and $\mathbf{S_2} \to \mathbf{S_3}$. Notice that unary performances are identical for $\mathbf{S_2}$ and $\mathbf{S_3}$ in all settings, but using the full system raises the recognition accuracy for $\mathbf{S_2}$ (since seeing the other heads allow to rule-out identities for the obfuscated head). We conclude that within events head obfuscation has only limited effectiveness, across events only blacking out all heads seems truly effective ($\mathbf{S_3}$ black).

Domain shift. In all scenarios, the recognition accuracy is significantly worse in the across events case than within events (about $\sim$50% drop in accuracy across all other dimensions). For a user, it is a better privacy policy to make sure no tagged heads exist for the same event, than blacking out all his heads in the event.

Qualitative result. In Figure 4.10 we show qualitative examples of recognition problems which have only been successfully solved by using a joint inference over non-tagged images. For example, second row in the "across events" block shows a face-obfuscated girl in bright pink clothing. Since the girl had always worn purple clothing in the tagged samples, the recognition system struggles linking this probe image to any of the tagged samples. However, when non-tagged yet face-visible instances are provided, the bright pink clothing links the probe and the non-tagged instances, and faces link the non-tagged and the tagged instances, eventually breaking the effect of face obfuscation in the probe image.

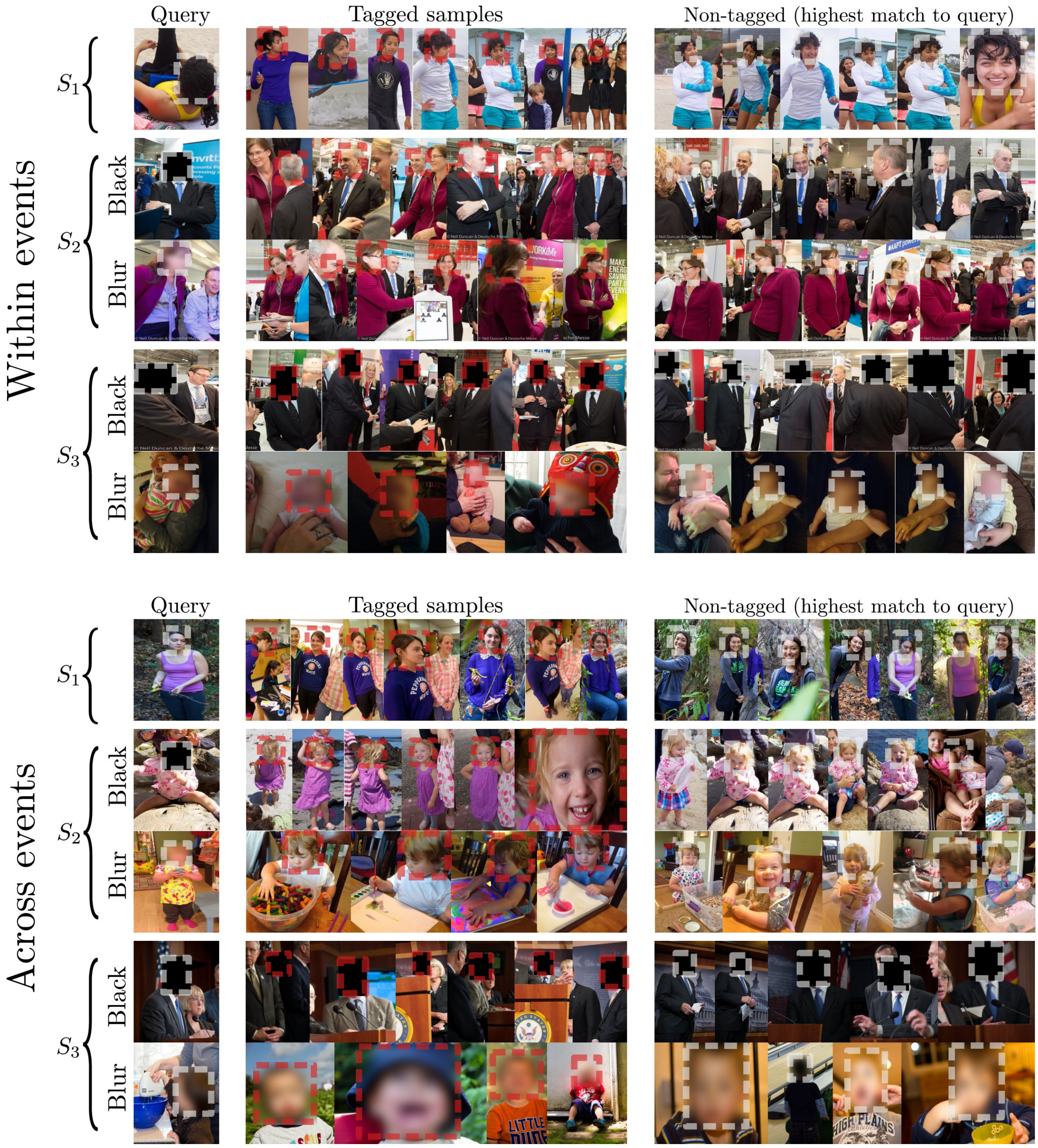


Figure 4.10: Examples of probe instances that are only successfully identified through a joint inference over multiple non-tagged samples. We order both tagged and non-tagged samples according to their predicted match against the probe.

4.5 DISCUSSION & CONCLUSION

Within the limitation of any study based on public data, we believe the results presented here are a fresh view on the capabilities of machine learning to enable person recognition in social media under adversarial condition. From a privacy perspective, the results presented here should raise concern. We show that, when using state of the art techniques, blurring a head has limited effect. We also show that only a handful of tagged heads are enough to enable recognition, even across different events (different day, clothes, poses, point of view). In the most aggressive scenario considered (all user heads blacked-out, tagged images from a different event), the recognition accuracy of our system is 12× higher than chance level. It is very probable that undisclosed systems similar to the ones described here already operate online. We believe it is the responsibility of the computer vision community to quantify, and disseminate the privacy implications of the images users share online. We conclude by discussing some future challenges and directions on privacy implications of social visual media.

Lower bound on privacy threat.. The current results focused singularly on the photo content itself and therefore a lower bound of the privacy implication of posting such photos. It remains as future work to explore an integrated system that will also exploit the images' meta-data (timestamp, geolocation, camera identifier, related user comments, etc.). In the context of the era of "selfie" photos, meta-data can be as effective as head tags. Younger users also tend to cross-post across multiple social media, and make a larger use of video (e.g. Vine). Using these data-form will require developing new techniques.

Training and test data bounds.. The performance of recent techniques of feature learning and inference are strongly coupled with the amount of available training data. May person recognition systems (Taigman *et al.*, 2014; Sun *et al.*, 2015; Schroff *et al.*, 2015) rely on undisclosed training data in the order of millions of training samples. Similarly, the evaluation of privacy issues in social networks requires access to sensitive data, which is often not available to the public research community (for good reasons (Narayanan and Shmatikov, 2010)). The used PIPA dataset (Zhang *et al.*, 2015b) serves as good proxy, but has its limitations. It is an emerging challenge to keep representative data in the public domain in order to model privacy implications of social media and keep up with the rapidly evolving technology that is enabled by such sources.

Part II

PRIVACY SOLUTION IN VISUAL DATA

The previous part was about analysis; this part is about aiming to provide solutions. Simple manipulation schemes such as face blurring are not effective solutions for privacy issues with state of the art person recognisers. What would be a better alternative? We propose two novel obfuscation techniques that are suprior in two aspects: obfuscation performance and image naturalness.

In Chapter 5 (Sun *et al.*, 2018), an inpainting-based identity obfuscation scheme is suggested. It leverages recent developments in generative models (GANs, Goodfellow *et al.* (2014)) to produce both natural and effective identity obfuscations.

Chapter 6 (Aditya *et al.*, 2016) is an interlude chapter. We present I-Pic technology, an image capturing framework that allows bystanders to obfuscate themselves in photos they appear. It is build on computer vision and cryptographic tools.

Included in the next part, Chapter 7 (Oh *et al.*, 2017c) proposes another obfuscation technique based on adversarial perturbations. This method results in a much more effective and nearly imperceptible changes on the input image compared to the obfuscation by inpainting. However, it requires a good knowledge of the target recogniser. We will discuss this work in the next part.

NATURAL AND EFFECTIVE OBFUSCATION BY HEAD INPAINTING 5

As we have seen in Chapter 4, blacking out or blurring head regions is ineffective against state of the art person recognisers. Nor do they result in natural-looking images. In this chapter, we propose a novel head inpainting obfuscation technique. Generating a realistic head inpainting in social media photos is challenging because subjects appear in diverse activities and head orientations. We thus split the task into two sub-tasks: (1) facial landmark generation from image context (e.g. body pose) for seamless hypothesis of sensible head pose, and (2) facial landmark conditioned head inpainting. We verify that our inpainting method generates realistic person images, while achieving superior obfuscation performance against automatic person recognisers.

The chapter is based on Sun *et al.* (2018). Dr Qianru Sun and Liqian Ma are the first authors of the paper (equal contribution). They have designed and trained the head inpainter module (§5.2). As a co-author, Seong Joon Oh has evaluated the obfuscation performance of the inpainted heads against our state of the art recogniser `naeil` (§5.3.6). Seong Joon Oh has also substantially contributed to the writing.

5.1 INTRODUCTION

Social media have brought about large-scale sharing of personal photos. While providing great user convenience, such a dissemination can pose privacy threats on users. It is essential to grant users an option to obfuscate themselves out of these photos. A good obfuscation method for social media photos should satisfy two criteria: *naturalness* and *effectiveness*. For example, putting a large black box over a person may be an effective obfuscation method, but would not be pleasant enough to share with friends.

Previous work on visual content obfuscation can be grouped into two categories: (1) *target-specific* and (2) *target-generic*. Some papers have proposed *target-specific* obfuscations, ones that are specialized against specific target machine systems, typically relying on adversarial examples (Oh *et al.*, 2017c; Sharif *et al.*, 2016). They yield nearly perfect identity protection with imperceptible changes on the input, but such a performance is guaranteed only against the targeted ones.

On the other hand, *target-generic* obfuscations change the actual appearance of the person such that generic classifier or even humans misjudge the identity. In its most crude form, commonly used obfuscation methods like black eye bar,

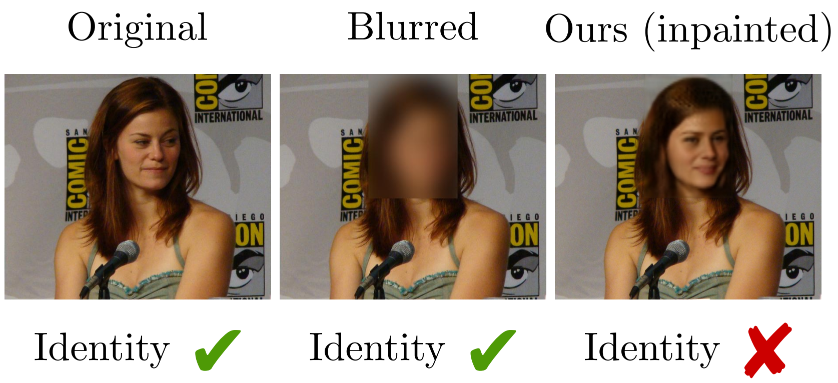


Figure 5.1: Our obfuscation method based on head inpainting generates much more natural patterns than common techniques like blurring, but still results in a more effective identity obfuscation against a recogniser.

face blurring, and blacking out head are examples of this type. These common patterns, unfortunately, are neither visually pleasant nor effective against machine systems (Oh *et al.*, 2016). This work proposes a *head inpainting* based approach to the target-generic identity obfuscation problem.

Generating realistic and seamless head inpainting on social media photos is hard. Subjects appear in diverse events and activities, resulting in varied backgrounds and head poses. Meanwhile, current generative face models are limited to frontal (Cole *et al.*, 2017) or strictly aligned faces (Lu *et al.*, 2017c).

We tackle the problem by factoring it into two stages. First, depending on the input, we detect or generate facial landmarks. In particular, when we have access to the original image, we detect facial landmarks. However, to keep our approach versatile, we also address the more challenging problem of generating facial landmarks from images that have been already obfuscated e.g. by blacking or blurring out the head region (called *blackhead* and *blurhead* in the remainder of the chapter, respectively). Then, conditioned on the face landmarks, we inpaint a realistic head that blends naturally into the context. We show that the resulting head-inpainted images mislead machine recognisers. Note that our method supports cases where the original face image is not available; existing head-obfuscated images on the web can be "upgraded" to our privacy enhanced head inpainting. Key contributions of the chapter are:

- Novel natural, effective obfuscation methods based on head inpainting.
- Novel landmark guided image generation approach for both head visible and blackhead cases in challenging social media photos.
- Novel facial landmark generator that effectively hypothesise realistic facial structures and poses given context in the scenario of blackhead.

5.1.1 Prior work on image inpainting

Image inpainting. In our work, we propose generative adversarial network (GAN) based method to complete head regions based on the context. Yeh *et al.* (2016) and Pathak *et al.* (2016) have also used GANs to generate missing visual contents, conditioning on the context. However, both approaches assume appearance and texture similarity between the missing part and the context. Our approach can generate head inpainting solely from body and scene context, without resorting to any information from the head region. In particular, unlike method in Yeh *et al.* (2016) which has been applied to aligned face images, our approach can be applied to challenging social media setup in which people appear with diverse poses and backgrounds by taking a two-stage approach.

Structure guided image generation. For generating realistic head inpainting that naturally blends into the given body pose and scene context, we have conditioned the inpainting on face landmarks. Some prior work has been devoted to the structure guided image generation; such a guidance has proved very helpful for generating images with complex inner structures (e.g. persons, (Ma *et al.*, 2017; Di *et al.*, 2017; Walker *et al.*, 2017; X and A, 2016; Ehsani *et al.*, 2018; Zhang *et al.*, 2017)). Ma *et al.* (2017) embed an arbitrary pose into a reference person image, and then refine the output by decoding more appearance details in the second stage. Di *et al.* (2017) use a similar structure embedding method to generate face image with detected facial landmarks on well-aligned face dataset. Walker *et al.* (2017) modelled the possible future movements of humans in the pose space, and then used the future poses generated as conditional information to a GAN to predict the future frames of the video. X and A (2016) propose to first generate a 3D surface normal map from a Gaussian signal and then synthesise images by painting style information on the map. Ehsani *et al.* (2018) solve the problem of object occlusion by first predicting the contour of invisible part then generate the appearance inside this contour. The second stage replies on the close visibility same as Context Encoder (Pathak *et al.*, 2016). Cole *et al.* (2017) recently introduce an approach face warping manipulation using landmark control on frontal face images. Unlike the above landmark work, our approach can not only generate new landmarks from body context, but also handle the large pose variances in Flickr images.

5.2 HEAD INPAINTING FRAMEWORK

We propose a context-driven head inpainting approach. We focus on social media photos which are challenging due to complex poses and scenarios. To learn an effective head generator from the data, we need strong guidance for which we use facial landmarks. Therefore, we factor the head inpainting task into two stages: landmark detection & generation and head inpainting conditioned on body context and landmarks.

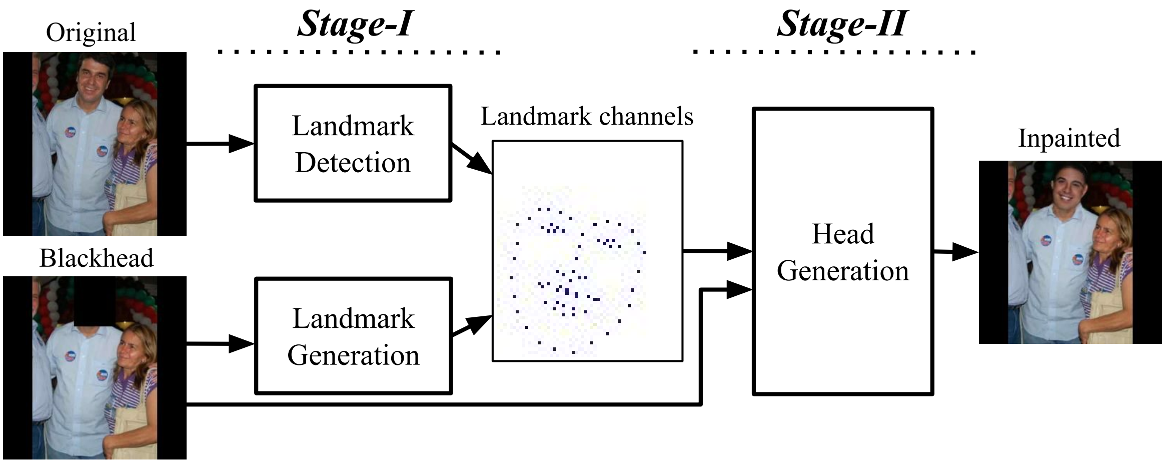


Figure 5.2: Our two-stage head inpainting framework. The input of stage-I is either the original or the blackhead image. The output is the inpainted image.

Figure 5.2 describes the global view of our two-stage approach. It takes either original (unobfuscated), blackhead, or blurhead image. We use blackhead image as the default example for obfuscated inputs. Given an original input, stage-I *detects* landmarks. However, when given an obfuscated input, stage-I *generates* landmarks. Stage-II takes an obfuscated image and the stage-I landmarks as input, and outputs the head inpainting.

5.2.1 Stage-I: landmark

The overview of stage-I is shown in Figure 5.3. For detecting landmarks on the original image, we use the detector in the python dlib toolbox (Kazemi and Sullivan, 2014). The output are 68 facial keypoints. For generating the landmarks on obfuscated images, we use a landmark generator network (G_L) trained adversarially with a discriminator (D_L).

Landmark generator (G_L). G_L has an autoencoder structure, and it contains two main parts: encoder and decoder. The encoder compresses the body/scene context of the blackhead image to a latent variable in the bottleneck layer which is then decoded to landmark coordinates by the decoder.

Encoder of G_L. The inputs to the encoder are the obfuscated image $\boldsymbol{I}$ and the head mask $\boldsymbol{M}$ corresponding to the head bounding box. As an output encoder yields a 32-dimensional latent vector $\boldsymbol{z}_L$. Encoder has an architecture consisting of 6 convolutional residual blocks.

Decoder of G_L. Taking $\boldsymbol{z}_L$ as input, the decoder generates 2×68-dimensional landmark coordinates $\boldsymbol{L}$. The decoder contains 6 fully connected residual blocks.

Training the encoder and decoder from scratch is challenging due to diverse body

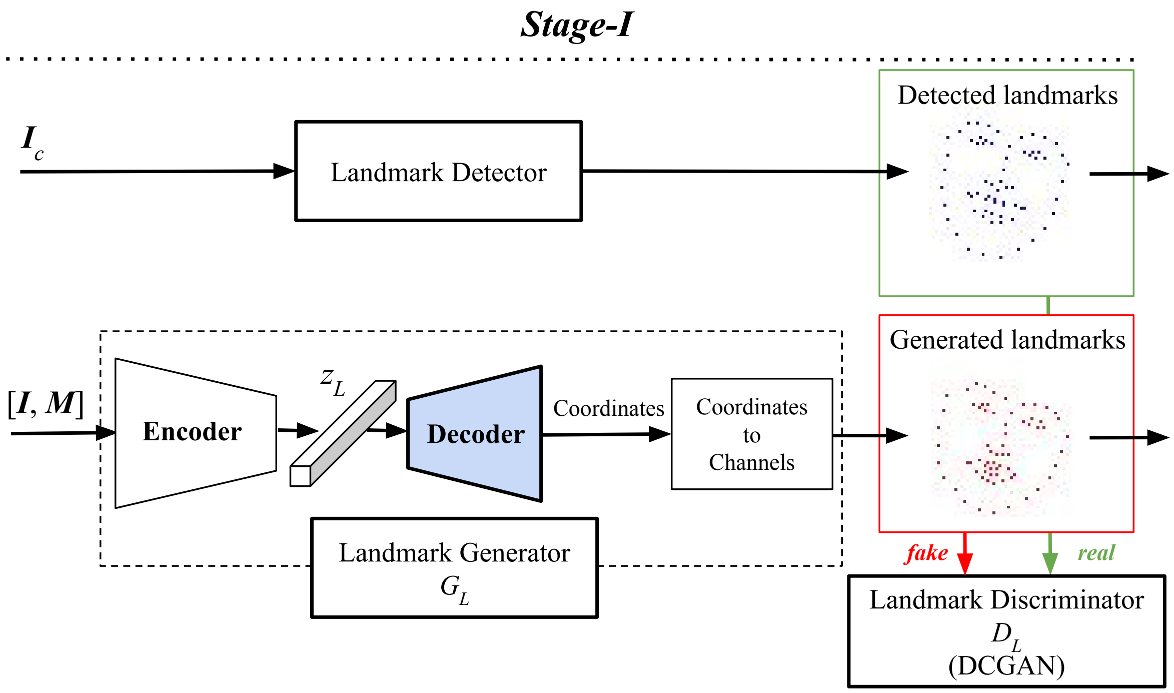


Figure 5.3: Stage-I: Landmark detection and generation. The detector takes the original image $\boldsymbol{I_c}$ as input, and the generator takes the obfuscated image $\boldsymbol{I}$ and the head mask $\boldsymbol{M}$ as inputs.

pose and background clutter in social media photos. Therefore, we consider first training a strong decoder and training the encoder from scratch with respect to the trained (and fixed) decoder. Such a procedure is inspired by the previous work on knowledge transfer between deep models trained on different tasks (Gupta *et al.*, 2016; Russakovsky *et al.*, 2015).

We consider training the decoder in three possible ways: (1) from scratch, (2) autoencoder, and (3) using the Point Distribution Model (PDM, (Cootes *et al.*, 1995)).

AE decoder (AEDec). The autoencoder reconstructs face landmarks using an encoder and a decoder through a bottleneck layer. Both are fully connected layers with ReLU activations. L_2 loss is used as the loss function.

PDM decoder (PDMDec). We consider using the Point Distribution Model (PDM) to better represent the 3D pose variations (Cootes *et al.*, 1995; Zadeh *et al.*, 2017). We train the PDM over the detected landmarks on PIPA *train* set images (Chapter 3). Our landmark points are parametrised using $\boldsymbol{p} = [s, R, t, q]$ denoting scale, orientation, translation and non-rigid transformations, respectively. The PDM decoder has the following formulation:

$$L = sR(\bar{L}_{3D} + \Phi q) + t \tag{5.1}$$

where $\bar{L}_{3D}$ denotes the mean value of the 3D landmarks mapped from our 2D data, and Φ the $3 \times n$ principal component matrix. The output $\boldsymbol{L}$ has $n + 6$ parameters. In the experiments we use $n = 34$ principal components.

Loss functions of G_L and D_L. We use the L_2 loss as well as an adversarial loss for optimisation. Landmarks trained only with the L_2 loss show noisy alignments; we found the adversarial loss to be useful at remedying this. We adopt the DCGAN discriminator (Radford *et al.*, 2015). The landmark coordinates are converted to channels to input to the convolutional layers, where the conversion process is differentiable. We have also tried a fully-connected discriminator, instead of the DCGAN discriminator, but the difference was marginal.

For training D_L, any landmark generated by G_L are labelled *fake*, while we use the *detected* landmarks as the *real* examples. Exact losses are formulated as follows:

$$\begin{aligned}\mathcal{L}_{D_L} =&\mathbb{E}_{\boldsymbol{X}\sim p_{data}(\boldsymbol{X})}\big[\log D_L(\boldsymbol{X})\big]+\\ &\mathbb{E}_{\boldsymbol{X}\sim p_{data}(\boldsymbol{X})}\big[\log\left(1-D_L(G_L(\boldsymbol{X}))\right)\big], \qquad (5.2)\\ \mathcal{L}_{G_L} =&\mathbb{E}_{\boldsymbol{X}\sim p_{data}(\boldsymbol{X})}\big[\log\left(D_L(G_L(\boldsymbol{X}))\right)\big]+\\ &\lambda_L\|G_L(\boldsymbol{X})-\boldsymbol{L}_d\|_2, \qquad (5.3)\end{aligned}$$

where $\boldsymbol{X}$ is the concatenation of the obfuscated image $\boldsymbol{I}$ (3 channels) and the head mask $\boldsymbol{M}$ (1 channel). L_d is the detected landmark coordinates. $\lambda_L \geq 0$ is a scalar weight.

5.2.2 Stage-II: inpainting

Stage-II generates the head inpainting based on the landmarks from Stage-I and the blackhead or blurhead image. Figure 5.4 shows an overview; the head generator G_H is trained adversarially with a head discriminator D_H.

Input. The 68-channel landmark heatmaps $\boldsymbol{L}_h$ from Stage-I are concatenated with the blackhead (or blurhead) image $\boldsymbol{I}$ as an input to the generator G_H. The landmark heatmaps provide the missing skeleton information in the obfuscated image.

We treat the blackhead image as *fake* and the original image as *real* the head discriminator D_H. Note that we use the whole body image instead of just head regions to provide sufficient information about the body and background to generate a realistic inpainting.

Head generator (G_H) and discriminator (D_H). The head generator G_H has a convolutional autoencoder with skip connections between encoder and decoder, inspired by the U-Net (Ronneberger *et al.*, 2015). The skip connections propagate image information directly from input to output, improving the fine-grained details in the output. The architecture of the head discriminator D_H is the DCGAN discriminator (Radford *et al.*, 2015).

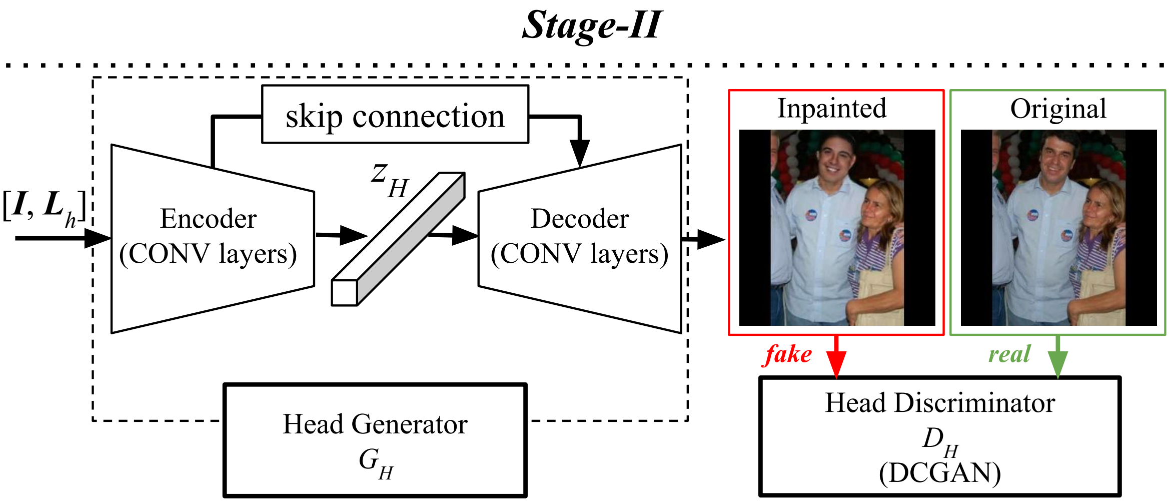


Figure 5.4: Stage-II: Head generation. The input are blackhead image $\boldsymbol{I}$ and landmark channels $\boldsymbol{L}_h$. The generator has an autoencoder structure which encodes the input to a bottleneck then decodes to a fake image. The discriminator is the same as in DCGAN (Radford *et al.*, 2015).

Loss function. We use the L1 and the adversarial losses to optimise G_H and D_H:

$$\begin{aligned} \mathcal{L}_{D_H} =& \mathbb{E}_{Y \sim p_{data}(Y)} \left[\log D_H(\boldsymbol{Y}) \right] + \\ & \mathbb{E}_{Y \sim p_{data}(Y)} \left[\log \left(1 - D_H(G_H(\boldsymbol{Y})) \right) \right], \end{aligned} \tag{5.4}$$

$$\begin{aligned} \mathcal{L}_{G_H} =& \mathbb{E}_{Y \sim p_{data}(Y)} \left[\log \left(D_H(G_H(\boldsymbol{Y})) \right) \right] + \\ & \lambda_H \| G_H(\boldsymbol{Y}) - \boldsymbol{I}_c \|_1, \end{aligned} \tag{5.5}$$

where $\boldsymbol{Y}$ is the concatenation of the obfuscated image $\boldsymbol{I}$ and the landmark heatmaps $\boldsymbol{L}_h$. $\boldsymbol{I}_c$ is the original image. $\lambda_H \geq 0$ is a scalar weight.

5.3 EXPERIMENTS

We evaluate the presented two-stage head inpainting pipeline on a social media dataset in terms of inpainting appearance and pose plausibility, as well as the identity obfuscation performance against machine recognisers. We analyse the impact of different input types (original, blackhead, and blurhead), different choices of landmark decoders, and the losses for the landmark generators (§5.2.1).

5.3.1 Dataset

We use the PIPA dataset (§3.2, (Zhang *et al.*, 2015b)), the largest social media dataset to date with people in diverse events, activities, and poses. It is a suitable for evaluating our methods under the social media obfuscation scenario.

In order to maximise the amount of training data, we have introduced a new partitioning of the images in PIPA. We partition the 2 356 identities into *train*+ set (2 099 identities, 46 576 instances) and *test*+ set (257 identities, 5 175 instances). We have further pruned both partitions with heavy profile or back-view heads, resulting in 34 383 instances in *train*+ and 1 909 in *test*+. The *train*+ set is used for training landmark and head generators; the *test*+ set is the evaluation set.

Our landmark and inpainting generators take a fixed-size image ($256 \times 256 \times 3$) as input. For every training and testing sample, we prepare the input by first obtaining the *body crop*, following the procedure in §3.4.2.

5.3.2 Scenarios and inputs

Our approach introduced in §5.2 is versatile and supports scenarios where the user (who wants to obfuscate an image) has access to the original image or only has access to already head-obfuscated images (e.g. blacked out). The necessity for this versatility is that social network service providers may aim to upgrade the privacy level by obfuscating images through blurring or blacking-out heads, even though it has been shown to be quite ineffective (Chapter 4).

In order to simulate multiple scenarios, we consider three types of inputs to our obfuscator: original, blackhead, or blurhead, where the latter two are common obfuscation techniques these days. We prepare blackhead and blurhead inputs following the procedure in Chapter 4. PIPA head box annotations indicate the head region to be obfuscated, which is either filled in with black pixels or smoothed with a Gaussian blur kernel specified in Chapter 4.

5.3.3 Quality of landmarks

Landmark detection or generation is the first stage of our inpainting pipeline. The landmarks should provide a plausible guess of the head pose and facial structure to guide a natural head inpainting in the next stage. In this section, we treat the detected landmarks as a good proxy to ground truth, and evaluate the landmark generation quality in terms of the deviation from the detected landmark. We measure the deviation via the mean of the L_2 distances between detected and generated landmark locations. The L_2 distances are normalised by the inter-ocular distances (Kazemi and Sullivan, 2014).

Note that the end goal of the landmark generator is not to replicate the detector, but only to provide a rough guidance for generating natural heads. In particular, small errors produced by the generator can even benefit us by inducing identity shift caused by a different facial structure. Hence, we only consider if the L_2 distance of the landmarks is within an acceptable range.

We investigate three axes of factors for our landmark generator. (1) Input type: original, blackhead, or blurhead. (2) Loss function: L_2 or $L_2 + D_L$ (adversarial loss). (3) Decoder type: trained from scratch (Scratch), autoencoder pretrained (AEDec), or Point Distribution Model pretrained (PDMDec). A summary of the quantitative

Obfuscation method			Evaluation									
	Landmark		Landmark		Inpainting		GoogleNet Acc. (%)			AlexNet Acc. (%)		
Input	Loss	Decoder	L_2	Norm. L_2	SSIM	mask-SSIM	h	h +b	h att.	h	h +b	h att.
Original	No head inpainting		/	/	1.000	1.000	85.6	88.3	72.2	81.6	85.3	66.0
Original	NN head copy-paste		/	/	0.872	0.195	1.2	7.1	67.5	1.4	6.1	46.2
Blur	No head inpainting		/	/	0.931	0.396	52.2	71.6	3.2	52.0	67.0	20.6
Blur	Detected landmarks		0.00	0.000	0.962	0.679	43.7	51.7	70.8	49.0	48.9	37.2
Blur	L_2	Scratch	6.32	0.230	0.954	0.578	36.2	48.4	66.8	44.6	44.6	36.7
Blur	L_2+D_L	Scratch	4.85	0.182	0.955	0.586	38.0	48.4	66.6	44.9	45.1	38.9
Blur	L_2+D_L	AEDec	4.77	0.180	0.951	0.585	37.5	48.0	66.1	43.9	45.0	37.5
Blur	L_2+D_L	PDMDec	4.50	0.168	0.953	0.593	37.9	49.1	66.7	45.1	45.6	38.0
Black	No head inpainting		/	/	0.000	0.000	2.1	67.0	14.0	2.1	63.2	1.7
Black	Detected landmarks		0.00	0.000	0.902	0.405	10.1	21.4	70.8	11.4	20.5	46.3
Black	NN landmarks		2.48	0.088	0.896	0.332	7.9	20.4	71.3	10.1	19.0	46.0
Black	L_2	Scratch	13.6	0.501	0.884	0.186	5.8	17.4	73.6	7.5	16.3	49.0
Black	L_2+D_L	Scratch	13.0	0.477	0.882	0.191	5.8	17.2	71.4	7.5	16.4	47.4
Black	L_2+D_L	AEDec	11.7	0.431	0.885	0.199	5.6	17.4	72.5	7.5	17.0	48.7
Black	L_2+D_L	PDMDec	12.3	0.453	0.885	0.196	5.6	17.4	71.0	7.4	16.6	51.2

Table 5.1: Evaluation of proposed obfuscation methods. We quantify the quality of the proposed obfuscation method against landmark quality, inpainting quality, as well as obfuscation effectiveness (person recognition rates). The head inpainter is always the G_H + D_H. We consider both GoogleNet-based and AlexNet-based recognisers. h and b indicate head- and body-cues for recognition. “h att.” refers to the attention on the head region for h +b.

results is given in Table 5.1 (“Landmark” column).

Input type. We compare the L_2 distance between generated and detected landmarks for three types of inputs: original, blackhead, or blurhead. For original images, we use detected landmarks, which gives by definition a zero L_2 distance. For reference, we measure the performance of our best landmark generator (will be discussed later) on the original images: 2.41 L_2 distance. This gives a lower bound (best case) on the L_2 distance for the generated landmark on obfuscated inputs.

We observe from Table 5.1 that blurhead inputs result in a better replication of the detected landmarks: 6.32 versus 13.6 for “Scratch” decoder with the L_2 loss. Blurhead images indeed contain structural information about the face keypoints. However, as we will see in the qualitative results (Figure 5.5), both result in pose-consistent landmark structures and natural heads.

Loss function. We compare two choices of the loss function: L_2 and $L_2 + D_L$. Given a blackhead input with the “Scratch” decoder, using only L_2 loss yields 13.6 distance from the detected landmarks. Adding the adversarial loss D_L improves the distance to 13.0. However, for blurhead input, the improvement due to the adversarial loss is much greater (from 6.32 to 4.85). It is thus advisable to use the adversarial loss for a better replication of the original landmarks.

Decoder. We consider three choices of the decoder in the generator G_L: learning from scratch (Scratch), pre-trained with autoencoder (AEDec), and pre-trained with PDM (PDMDec). For both blurhead or blackhead cases, conditioning the decoder with either AEDec or PDMDec helps generating landmarks closer to the detected ones: for blackhead input, L_2 distance metric improved from 13.0 to 11.7 and 12.3, respectively for AEDec and PDMDec.

5.3.4 Head inpainting quality

We evaluate the quality of generated heads. As a proxy to the naturalness of the output, we use a perceptual metric (SSIM, (Wang *et al.*, 2004)) to measure the perceptual distance to the original images. For measuring the head region quality only, we use the mask-SSIM (Ma *et al.*, 2017).

As a baseline, we consider the nearest neighbour method: search for the nearest neighbour head in the training data based on the L_2 distance of the detected landmarks, and replace the head patch. This ignores the blending with surroundings, resulting in unpleasant output and a low SSIM score 0.872.

We note that the blackhead based head inpainting results in significantly lower SSIM measures than the blurhead based versions: 0.902 versus 0.962 SSIM and 0.679 versus 0.405 mask-SSIM for the detected landmarks case.

5.3.5 Qualitative examples

While quantitative measures are useful for summarising the trend, they can be misleading especially when measuring quantities that are hard to measure - e.g. naturalness. We present qualitative examples of the generated landmarks and head inpainting to show that our pipeline generates high-quality head images that blend naturally into the background and body pose. See Figure 5.5 for the qualitative examples.

Detected versus generated landmarks. We compare the quality of detected and generated landmarks and the corresponding inpainted heads in Figure 5.5. The detected landmarks closely follow the original landmarks, while the generated landmarks, especially for blackhead cases, result in landmarks (and thus inpainting) with different head poses, explaining the greater L_2 distances shown by blackhead landmarks. However, it is important to note that the generated landmarks are still plausible with respect to the body pose and activity. Comparing the head inpainting, we observe again better naturalness in the blurhead-based inpainting than in the blackhead-based ones. However, the final output in all the cases are not implausible.

Blackhead versus blurhead. Columns 2,4 and columns 3,5 in Figure 5.5 show respective examples for blurhead- and blackhead-based landmarks and the corresponding inpainted heads. Involving blurred head images during landmark and head generation results in inpainting that resembles the original head, especially the head pose and hair colour/style (e.g. ID 690). On the other hand, not providing any information in the head region results in a significantly different, yet plausible, head images. In particular, when landmarks are generated, the resulting head images are drastically different from the original one.

5.3.6 Evaluation of obfuscation performance

While on the one hand, a good obfuscation scheme should produce plausible head replacement, it is also crucial for the output to disable the target recognition system from correct predictions. In this section, we quantify the obfuscation success rate against the person recogniser models (`naeil`) from Chapter 3. Unlike typical face recognisers, `naeil` uses body cues for recognition as well, resulting in ineffectiveness of blurring or blacking out heads (Chapter 4). In this work, we show that head replacement is a more effective obfuscation scheme than simple head blurring or blacking out. In our work, we use both AlexNet- and GoogleNet-based `naeil` with features from either head (`h`) or head and body together (`h` +`b`).

Head inpainting provides good protection. Table 5.1 shows the obfuscation performance (columns `h` and `h` +`b`). Under no obfuscation, GoogleNet `h` +`b` recognition performance is 88.3%. Black/blurring baselines give 67.0%, and 71.6%, respectively – confirming the observation in Chapter 4 that these are ineffective. On the other hand,

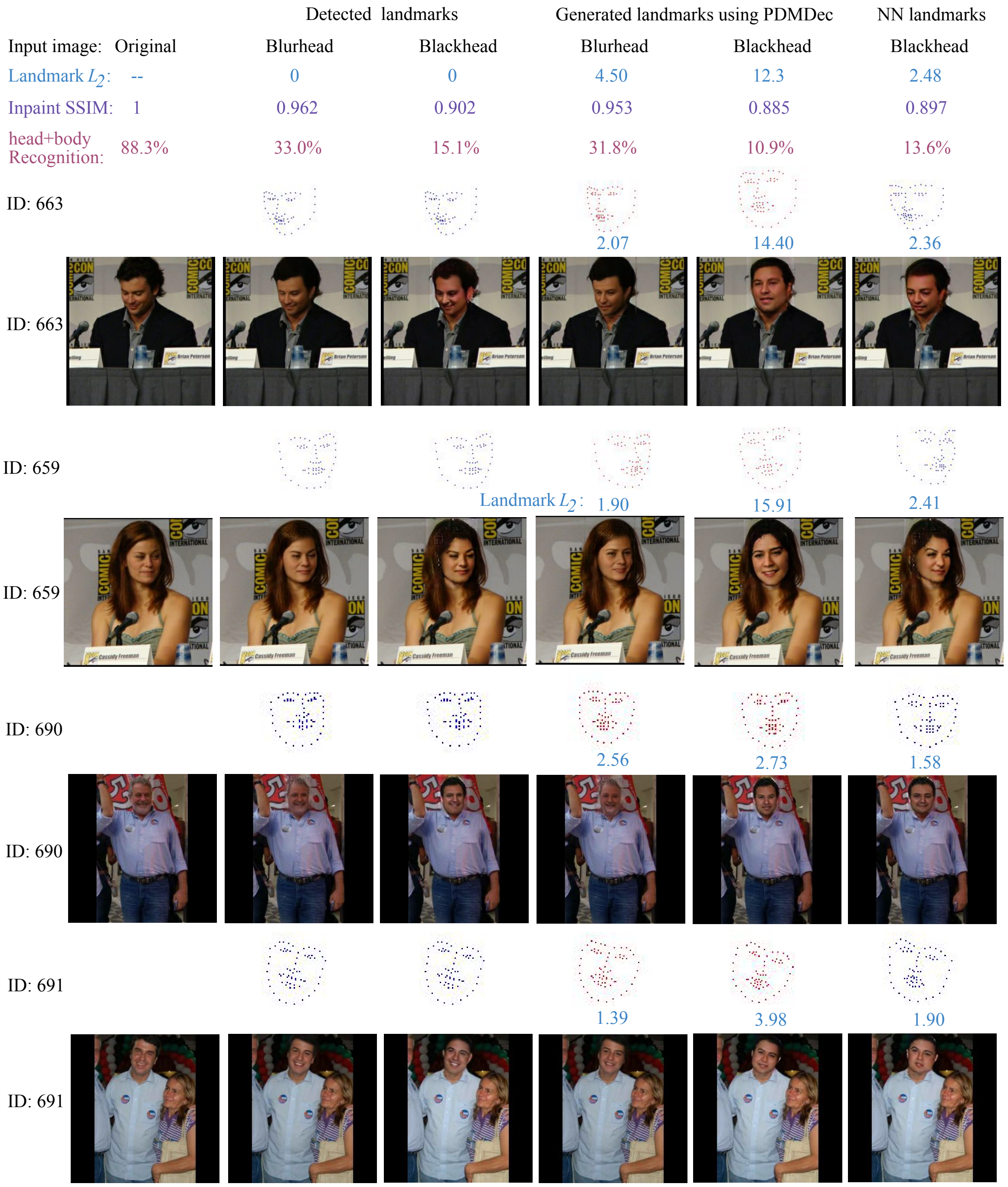


Figure 5.5: Visualization results. We show the head inpainting results using detected and generated landmarks (from the PDMDec model). Top rows present key quantitative numbers for reference. Landmark generation error (distance to the detected one) is also given for each single instance.

our head inpainting methods result in $< 50\%$ (blurhead input, generated landmarks) and $< 18\%$ (blackhead input, generated landmarks) recognition rates for GoogleNet `h` +`b`. They are more effective protection techniques than blacking or blurring head regions.

Cues used. We compare the recognition rates between `h` and `h` +`b`. When the recogniser relies solely on head cues, while the head has been inpainted, then the recognition rates are lower than the `h` +`b` counterparts. For example, the last row method against `h` recogniser gives 5.6% versus 17.4% for `h` +`b` (GoogleNet), nearly reaching the chance level recognition rate (2.1%).

Input type. While having access to blurhead images help generating more plausible landmarks as well as visually natural head inpainting, they may leak identity information. We compare the recognition rates when blurhead or blackhead inputs are used. Our head inpainting based on the blackhead result in $17\% \sim 18\%$ accuracy (GoogleNet), while blurhead based results are in the range $48\% \sim 50\%$ accuracy (GoogleNet). This confirms that indeed there exists a trade-off between plausibility of generated heads and the obfuscation performance.

Detected vs generated landmarks. While identity information may leak through blurred heads, it may also leak through the landmark detections (face shape). On the other hand, generated landmarks enjoy the possibility to come up with an equally plausible landmark hypothesis but with different face shapes. For the blackhead input, the detected landmarks indeed result in higher recognition rate (21.4%) than generated ones (e.g. 17.4%, last row), with similar trend for the blurhead cases (GoogleNet).

Rationale for good obfuscation – recogniser attention. We have verified that our head obfuscation scheme exhibits better performance than commonly used ones like blacking and blurring. We give a rationale for this phenomenon by means of the *recogniser attention*. Given an input, *recogniser attention* refers to the regions in the image where the recogniser extracts cues from. We hypothesise that while blacked or blurred heads induce recogniser attention on non-head regions, our inpainted heads attract attention on the heads due to the realism of the inpainted heads.

For the *recogniser attention* we have used the gradient-based mechanism from Simonyan *et al.* (2014). We first compute the gradient of the neural network prediction with respect to the input image; take maximal absolute values along the RGB channel; and then smooth with Gaussian blurring. To quantify the chance of attending on the head region, we have computed the "head contribution" score by estimating

$$\text{head contrib.} = \mathbb{P}[\text{max attention is inside head region}]$$

over the test samples.

See the "`h` con." columns of Table 5.1 for the results. We observe that while the original image has 72.2% chance of inducing attention on the head region, blacked or blurred heads are much less likely to attract the recognisers' attention (14.0% and 3.2%, respectively). This explains why `h +b` is still performing well: it simply ignores the confusing head cue. On the other hand, our inpainted heads still attract the recognisers' attention as much as non-obfuscated head images do (71.0% versus 72.2%). This indicates that the realism of inpainted heads encourages the recogniser to rely its decision on the inpainted head, effectively leading to its misjudgement.

On the ethical issue of predicting other identities. Ethical problems might entail if the obfuscation mislead the recogniser into confidently predicting other identities in the gallery set. We have measured the SVM prediction confidence (1-vs-all SVM) on the original as well as obfuscated images to ensure that the obfuscation results in a uniformly low prediction scores.

On the original images, the argmax identity is predicted with SVM score 0.63 on average. On the other hand, our inpainting obfuscation conditioned on blurhead results in -0.29 average SVM score for the argmax prediction, while conditioning on blackhead results in much lower maximal SVM score of -0.52. This confirms that the inpainting based obfuscation does not shift the identity prediction to another person with high confidence. If one filters out predictions with low confidence scores, a common practice in application, then the head-inpainted images will most likely be filtered out as "no confident prediction".

Sensitivity to recognition systems. We show the impact of the recognition system on the obfuscation performance. Our obfuscation approach is *target-generic*: it is not generated with respect to a particular recognition system and is expected to work against a generic recogniser. To show this, we present a comparison of obfuscation performance against GoogleNet and AlexNet based recognisers. Note that they result in very different recognition systems in terms of the depth and the number of parameters in the architecture.

In Table 5.1, we reach the same conclusion regarding the efficacy of our inpainting-based obfuscation. Our inpainting method "$L_2 + D_L$ - PDMDec" decreases the recognition rate of AlexNet-based `h +b` from 67.0% to 45.6% (blurheads) and from 63.2% to 16.6% (blackheads). We again observe that the contribution from head region increases as our method inpaints realistic head images. Through results on two very different recognisers, we confirm that our method is indeed target-generic.

5.4 CONCLUSION

To address the problem of obfuscating identities in social media photos, we have presented a two-stage head inpainting method. Although the social media setup is more challenging than previous face-generation setups (diverse head, body poses, and backgrounds), our method has proved to generate natural obfuscation patterns that effectively confuses an automatic person recogniser. In particular, our method is *target-generic*: the obfuscation is designed to work against any recogniser, be it human or machine. Also, the method does not require access to the original image, enabling an “upgrade” scheme for existing weak obfuscation patterns (e.g. head blurring or blacking out).

6 I-PIC: A PLATFORM FOR PRIVACY-COMPLIANT IMAGE CAPTURE

This chapter is an interlude chapter that introduces an orthogonal technology where obfuscation methods could be plugged in to enhance user experience. We present I-Pic, a trusted photo-capturing software platform that allows bystanders in photographs to obfuscate themselves according to their privacy preferences. In I-Pic, users choose a level of privacy (e.g., image capture allowed or not) based upon social context (e.g., out in public vs. with friends vs. at workplace). Privacy choices of nearby users are advertised via short-range radio, and I-Pic-compliant capture platforms generate edited media to conform to privacy choices of image subjects.

I-Pic uses secure multiparty computation to ensure that users' visual features and privacy choices are not revealed publicly, regardless of whether they are the subjects of an image capture. Just as importantly, I-Pic preserves the ease-of-use and spontaneous nature of capture and sharing between trusted users. Our evaluation of I-Pic shows that a practical, energy-efficient system that conforms to the privacy choices of many users within a scene can be built and deployed using current hardware.

The chapter is based on Aditya *et al.* (2016). Dr Paarijaat Aditya was the first author of the paper. He has developed the overall I-Pic architecture and written the major part of the manuscript. As a co-author, Seong Joon Oh has participated in the computation of image features required in the I-Pic pipeline (§6.4.1).

6.1 INTRODUCTION

The spontaneity afforded by mobile devices with cameras have led to new creative outlets that continue to have broad and lasting social impact. As every facet of event reporting, ranging from personal journals to war correspondence, is transformed, however, there is a growing unease about the dilution of privacy that inevitably accompanies digital capture in public, and in some cases, private fora. This work describes I-Pic, a platform for *policy-compliant* image capture, whereby captured images are automatically edited according to the privacy choices of individuals photographed. I-Pic's design was motivated by a user-study, described in Section 6.2, which found that:

Capture policies should be individualised. Privacy concerns vary between individuals. Even in the same situation, different subjects have different preferences. This finding motivated I-Pic to preclude options that impose blanket or venue specific policies (Roesner *et al.*, 2014; Raval *et al.*, 2016)[5].

Policies should be situational. Study subjects stated consent to be photographed at certain times, places, events, or by certain photographers, but would make different choices in other circumstances. This motivated I-Pic to not impose a static policy per individual (Bo *et al.*, 2014), and to avoid solutions that require prior arrangements between specific subjects and photographers (whitelisting or blacklisting).

Compliance by courtesy is sufficient. An overwhelming majority of our subjects stated that they would choose to comply with the privacy preferences of friends and strangers, especially if doing so didn't interfere with the spontaneity of image capture. I-Pic provides such a platform but is not meant to stop determined users from taking pictures against the wishes of others; indeed, these users could simply use a non-I-Pic compliant device.

Consider a strawman system where mobile devices broadcast their owner's privacy preferences via Bluetooth. Without additional information, a camera would have to edit the image according to the most restrictive policy received, even if the corresponding person does not appear in the image at all. To be practical, polices must be accompanied by a visual signature so that a camera can associate a person captured in an image with a policy.

However, Bluetooth transmissions can cross walls, which would create a serious privacy problem if visual signatures were broadcast in the clear: Next-door neighbours could identify persons whom they have never seen or photographed. To avoid this problem, I-Pic relies on secure multiparty computation (MPC) to ensure that a capture device learns only a person's privacy choice, and only if that person was captured; otherwise, neither side learns anything.

User studies and privacy requirements inform the architectural components of I-Pic: Users advertise their presence over BLE (Bluetooth Low Energy): these broadcasts are received by I-Pic-compliant capture platforms. When an image is taken, the platform determines if any of the captured people match the visual signatures of nearby users using MPC. If there is a match, the platform learns the policy and edits the image accordingly, e.g., by occluding the person's face. To maintain the responsiveness of image capture, unedited images are shown to the photographer immediately, but cannot be shared until the image is processed in the background.

After presenting the results of our online survey in §6.2, we describe the main technical design of I-Pic in §6.4, along with prior work in face recognition and cryptography we build on. Next, we presents results of an experimental evaluation in §6.5. We conclude in §6.6.

[5] Lost Lake Cafe, Seattle restaurant, kicks out patron for wearing Google Glass, https://www.huffingtonpost.com/2013/11/27/lost-lake-cafe-google-glass_n_4350039.html

6.1.1 Related work

Privacy in the presence of recording devices. Hoyle *et al.* (2014) seek to understand users' concerns about continuous recording using wearable cameras, by studying a large user population of avid life-loggers. Denning *et al.* (2014) conduct a large scale user survey to understand bystanders' privacy concerns in public places like coffee shops and possible ways to mitigate them. Our online survey additionally shows that privacy concerns are very personal and dependent on the situation.

Roesner *et al.* (2014) present a system that shares a venue's privacy preferences with wearable devices in an unobtrusive way. The idea is to convey privacy expectations associated with places like gyms and washrooms with broadcast messages or visual signs. The wearable devices in the venue pick up these messages or visual cues and obey the specified privacy protocol. Unlike I-Pic, this system has no way to associate a privacy policy with an object or person that appears in an audiovisual recording.

Visual markers to convey privacy policies to nearby wearable recording devices are also used in Raval *et al.* (2016). Jung and Philipose (2014) explore the expression of bystanders' privacy intent using gestures. Unlike I-Pic, these approaches require either physical tagging of objects and locations, or explicit user actions (i.e., gestures) to convey privacy choices. Moreover, I-Pic enables user-defined, personalised, context-dependent privacy choices.

In the work by Bo *et al.* (2014), individuals wear clothes with a printed barcode, which encodes the wearer's public key. When an image of an individual showing face and barcode is uploaded to an image server, the server garbles the face pixels, using the public key encoded in the barcode. Only the individual who owns the associated private key can later extract the actual face image. I-Pic, on the other hand, does not require its users to wear any visual markers, it does not require users to trust an image server with their private images, and can support context-dependent privacy policies.

In D'Antoni *et al.* (2013); Jana *et al.* (2013b,a), the authors address privacy concerns in untrusted perceptual and augmented reality applications, by partially processing media stream within the trusted platform, thus denying apps access to the raw media streams. An augmented reality app, for instance, might be provided only with the position of relevant objects within a video stream sufficient for the app to overlay its own information, but not the full video. I-Pic also relies on the trusted platform, but focuses on enforcing individual's privacy policies regarding image capture by nearby devices.

Zero-Effort Payments (Smowton *et al.*, 2014), similar to I-Pic, uses face recognition and proximate device detection using BLE to identify a user in an image, but their goal instead is to create a mobile payment system. Unlike I-Pic, which is tuned to identify even small faces in diverse range of photographic contexts, their system is meant to visually identify a user, with human assistance, when she is in close proximity to the cashier. Furthermore, they acknowledge concerns of user privacy in such a monitored environment and propose the use of signage indicating that a face recognition system is deployed in the area. Such a privacy solution is only viable in select scenarios, and lacks the flexibility provided by I-Pic.

Visual fingerprints. Performance on human identification and re-identification tasks has greatly improved over the last decade. Most notably, face recognition on large databases in realistic settings is even approaching human performance (Taigman *et al.*, 2014). Besides the identity, a person can also be described and identified by a set of attributes (Bourdev *et al.*, 2011; Zhang *et al.*, 2014). I-Pic uses a state of the art face recognition algorithm based on neural networks, but can benefit from using semantic attributes describing a face, including features from other body parts in addition to the face.

Cryptographic primitives. There is complementary work to protect the privacy of biometric data (Lingli and Jianghuang, 2010; Wang and Plataniotis, 2010) by projecting or encrypting representations. It is possible that these approaches could be used in I-Pic to further reduce trust in the Cloud service by obscuring users' visual signatures.

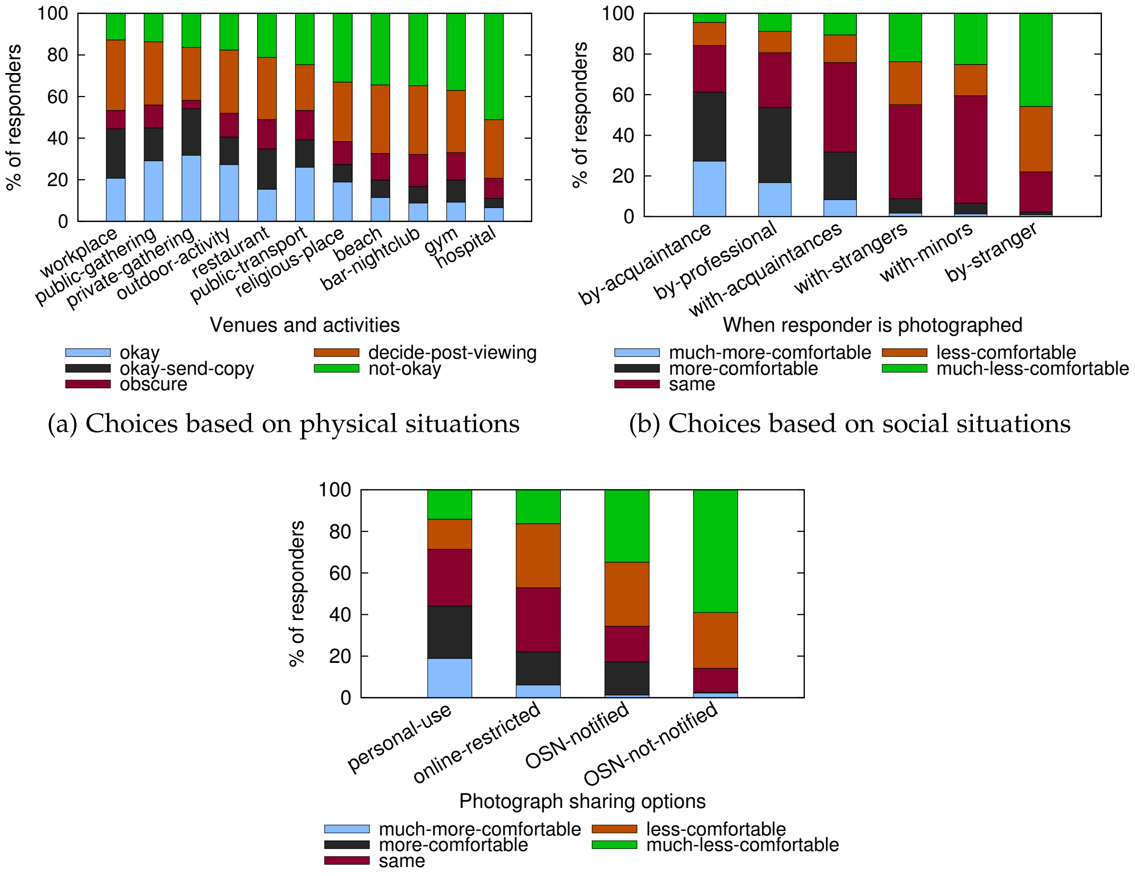
InnerCircle (Hallgren *et al.*, 2015) describes a secure multi-party protocol for location privacy, which computes in a single round whether the distance between two encrypted coordinates is within some radius r. This computation is similar to I-Pic's secure dot product and thresholding computation. However, the protocol's efficiency degrades exponentially with the number of bits of precision of the distance. Since our threshold comparison involves dot products of large feature vectors, we use garbled circuits for the threshold comparison instead.

6.2 ONLINE SURVEY

I-Pic's design was informed by an online survey designed to provide a broader perspective on personal expectations and desires for privacy. The survey, and experiments with I-Pic, were conducted with user consent under an IRB approval from the University of Maryland. The survey included an optional section on user demographic, including gender, age, and ethnicity.

We publicised the survey on mailing lists and online social networks on November 10th, 2015. The survey is available online at http://goo.gl/forms/6tGGoYmFFG, and the results here present a snapshot of all responses collected on December 4th, 2015. As of this date, there were 227 responses, with 208 responders also answering the demographic questions. Respondents represented 32 countries. The age distribution is shown in Table 6.1.

Questions in the survey envisioned different venues and activities and presented participants with different privacy options: (a) agree to be captured in any photograph, (b) agree, but would like a copy of the image, (c) please obscure my appearance in any image, (d) can decide my preference only after viewing the photo, or (e) do not wish to be captured in any photograph. Participants were asked



(a) Choices based on physical situations

(b) Choices based on social situations

(c) Choices based on use of images

Figure 6.1: Variety in privacy preferences under similar physical, social and image usage scenarios

to choose the privacy action they considered most appropriate for each scenario (Figure 6.1(a)). To help visualise a common scenario and to provide perspective for others, participants were shown an image of people on a platform waiting to board a train, some with faces clearly visible. The survey also gauged individual's level of comfort depending on their relationship to the photographer or the other subjects in the photograph (Figure 6.1(b)). Finally, we asked how potential uses of an image influence responders' level of comfort with being captured (Figure 6.1(c)).

In Figure 6.1(a), the x-axis is sorted by the percentage of responders who chose the most private action of "do not wish to be captured", increasing from left to right. Results show a mix of privacy concerns for different scenarios. In Figures 6.1(b) and 6.1(c), the x-axis is sorted by the percentage of responders who were less comfortable with photography, increasing from left to right. Again, for these social situations or image usage scenarios, the privacy concerns of responders is not uniform. *These results demonstrate the necessity of diversity in privacy policy, and argue against venue based policies (Roesner et al., 2014; Raval et al., 2014).*

Age group	Fraction of participants
less than 20 years	9.2%
20 - 30 years	56.6%
30 - 40 years	25.1%
40 - 50	4.8%
more than 50 years	3.9%
Unspecified	0.4%

Table 6.1: Age groups of survey participants

Number of privacy preferences	Fraction of participants
1	12.7%
2	27.8%
3	32.2%
4	19.4%
5	7.9%

Table 6.2: Variety in privacy preferences for same person

Unsurprisingly, privacy preferences are not unanimous for any scenario; there are, however, trends. Responders tend to be more restrictive in venues such as beaches, gyms and hospitals (in Figure 6.1(a)); with strangers in a social situation (in Figure 6.1(b)); and when images can potentially be shared online (in Figure 6.1(c)). These trends can be useful as they suggest default policies appropriate for different situations.

Table 6.2 shows the percentage of responders versus the number of different privacy choices for each responder. The table shows that individuals prefer different privacy choices depending on the given situation. *This finding illustrates the utility of context-specific policies, and demonstrates the shortcomings of individualised hard-coded policies, e.g., bar-codes on clothing (Bo et al., 2014).*

The survey asked whether responders cared about *by-stander* privacy when respondents themselves capture images. An overwhelming majority (96.47%) answered in the affirmative, motivating a system such as I-Pic. About a quarter (28%) agreed if the overhead of the solution was low; another quarter (26%) agreed if the aesthetics of images remain good.

Respondent Selection Bias. The survey was voluntary and anonymous. The URL for the survey was advertised on mailing lists and social networks used by the authors and their friends, leading to a bias in how respondents learned about the survey. However, we believe that the results presented here still have merit as they represent views across different age groups and ethnicities. The results overwhelmingly support the thesis that users often desire privacy from digital capture in social situations, and further that "one-size-fits-all" solutions to image privacy are not effective. Moreover, as photographers, the responders overwhelmingly consider bystander privacy to be important. These observations inform I-Pic's design, described next.

6.3 I-PIC ARCHITECTURE

Figure 6.2 shows I-Pic's major components and their interaction. The two types of principals in the system are *bystanders* or users who may be photographed, and *photographers* who capture images. Both are assumed to operate an I-Pic-compliant *platform*. Associated with each principal is a cloud-based *agent* to which the principals offload compute-intensive tasks. The photographer is associated with a *Capture Agent*; each bystander is associated with a *Bystander Agent*. We note that agents are logical constructs; functions provided by the agent can be implemented within mobile devices should I-Pic be used without wide-area connectivity.

I-Pic requires a one-time *Association* protocol between users and their agent. Users *periodically broadcast* their presence using BLE. Once an image is captured, the *Face Detection*, *Feature Extraction*, and *Secure Matching* protocols are executed. If a user is identified, the capture platform uses the *Policy Enforcement* protocol to modify the photograph as requested. We describe these sub-protocols next.

Association. Users select an agent as a proxy and provide it with photographs, which are used to train an SVM classifier for face recognition. A user trusts her agent not to leak her visual signature. The association protocol also exchanges a master key between agent and user's device, which is used to generate session keys in the future. Next, users initialise their privacy profile, which is locally stored on their device, by choosing relevant contexts based on location (e.g. office, home, gym, bar/restaurant, public spaces) and time (work hours, off-work hours), and by choosing an appropriate action for each context (agree to appear with face, blur face).

Periodic broadcast. Users periodically broadcast a encrypted policy that specifies how to treat the user's picture if she appears in a photograph. This broadcast also includes sufficient information to identify the user's agent. The policy is encrypted with a session key generated using the current time (divided into 15-minutes epochs) and the master key exchanged with the user's agent. Capture platforms receive and cache policies. Once a photograph is captured, if a user is identified, then the associated policy can be decrypted.

Secure matching. Upon image capture, the platform detects and tries to recognise faces. These components leverage our prior work in face detection (Mathias *et al.*, 2014) and facial feature extraction (Chapter 3), as detailed in §6.4.1. The capture platform encrypts the extracted features and uploads them to its agent, along with the network identifiers of all bystander agents that it has received as broadcast

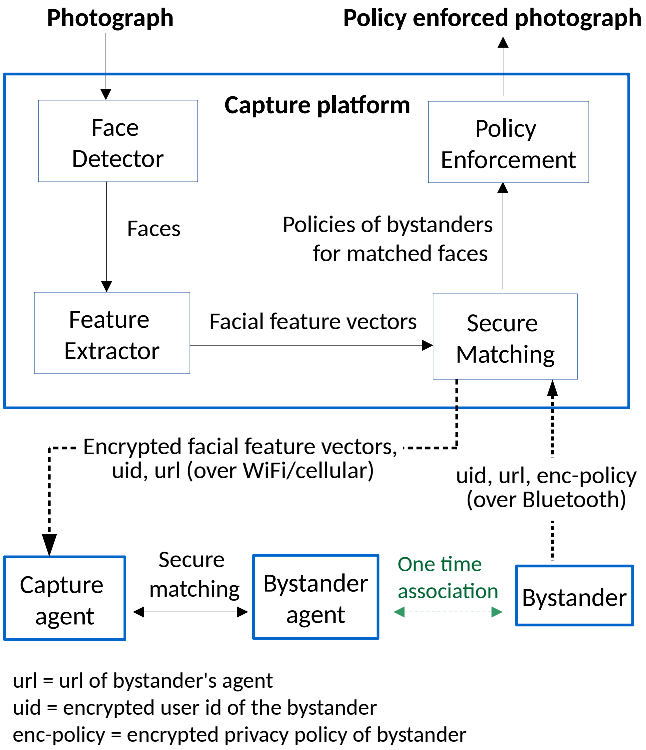

Photograph
Policy enforced photograph
Capture platform
Face Detector
Policy Enforcement
Faces
Policies of bystanders for matched faces
Feature Extractor
Facial feature vectors
Secure Matching
Encrypted facial feature vectors, uid, url (over WiFi/cellular)
uid, url, enc-policy (over Bluetooth)
Capture agent
Secure matching
Bystander agent
One time association
Bystander
url = url of bystander's agent
uid = encrypted user id of the bystander
enc-policy = encrypted privacy policy of bystander

Figure 6.2: I-Pic major components

recently. The *Capture Agent* and the *Bystander Agent* compare extracted features and a bystander's classifier weight vector by implementing a secure dot-product protocol (Goethals *et al.*, 2004) followed by a secure threshold comparison protocol based on garbled circuits (Yao, 1986). If the threshold passes, then the session key used to encrypt user's policy is revealed to the capture platform.

Policy enforcement. When granted a session key for a user, the capture platform decrypts the corresponding user's privacy policy and performs the action requested. Our current implementation only supports face obfuscation, which we implement using the OpenCV library. More sophisticated techniques exist. For instance, it is possible to replace the face with another hypothetical, non-existent one (Chapter 5) or introduce unnoticeable perturbations on the image to defend against specific recognition systems (Chapter 7). While such advanced image processing techniques are not the subject of this chapter, I-Pic can take advantage of them. If a captured face cannot be matched against any bystander, but all advertised policies have been evaluated, I-Pic defaults to blurring the face. This protects the privacy of bystanders who either do not own a smart device or are not I-Pic users. Similarly, all unmatched faces are blurred if the identification protocol does not complete for some policies, likely due to lack of network connectivity. The platform maintains an encrypted copy of the original image, which can be used to release an unblurred face in the original image as the protocol completes in the future.

6.3.1 Threat model

I-Pic's cryptographic protocols ensure that a non-compliant capture device cannot learn the feature vectors of a bystander who does not appear in a captured image. For privacy policies of bystanders to be correctly applied, the capture platform on users' devices is assumed to implement the I-Pic protocol correctly. Third-party applications installed on users' devices are untrusted.

Users of capture devices may be able to bypass I-Pic by "rooting" their device; a different implementation could integrate I-Pic into the device firmware or implement the protocol on a trusted hardware platform, thus raising the bar for bypassing I-Pic's privacy protection. We dismissed this approach, because uncooperative photographers could in any case use a non-I-Pic compliant camera. Our goal instead is to enable cooperative photographers to respect bystander's privacy wishes in an unobtrusive manner, without introducing new attack vectors. We believe that most users welcome the ability to automatically comply with bystander's wishes, as it enables them to take pictures freely, without worrying whether they might offend others. This was also observed in our online survey (§6.2), where 96% of the participants indicated that they cared about bystanders' privacy.

The *Bystander Agent* must be trusted by the bystander not to leak her visual signature. The *Capture Agent*, on the other hand, does not have access to either the users' visual signature stored on the *Bystander Agent* or the features vectors extracted by the capture device. However, *Bystander Agent* and *Capture Agent* are assumed not to collude, else they could jointly extract the feature vectors of people captured in an image. *Capture Agent* is additionally expected to construct the garbled circuit used for secure threshold comparison (described in §6.4.2) accurately.

Cloud agents learn when an I-Pic compliant device captures an image, and the *Capture Agent* learns the IP address of that camera device (Technically, both could be spoofed since the request may use an identifier without capturing an image, and the source IP address in a request could be that of a forwarding relay). I-Pic protocols are designed to ensure that the cloud agents do not learn if a user appears in an image, or the user's current context or policy. The following §6.4 describes the I-Pic protocols in detail.

6.4 I-PIC DESIGN

We describe the design of I-Pic in more detail. Figure 6.3 shows the I-Pic workflow in normal operation.

I-Pic compliant devices broadcast their encrypted $(userid, policy)$ pairs periodically. They additionally discover other Bluetooth devices periodically and add any received pairs to a local cache of nearby users. The entries are flushed from the cache when a device's broadcast has not been received for 10 minutes.

When an image is captured, I-Pic intercepts the raw image data. The captured image is available for viewing immediately but cannot be shared until the image

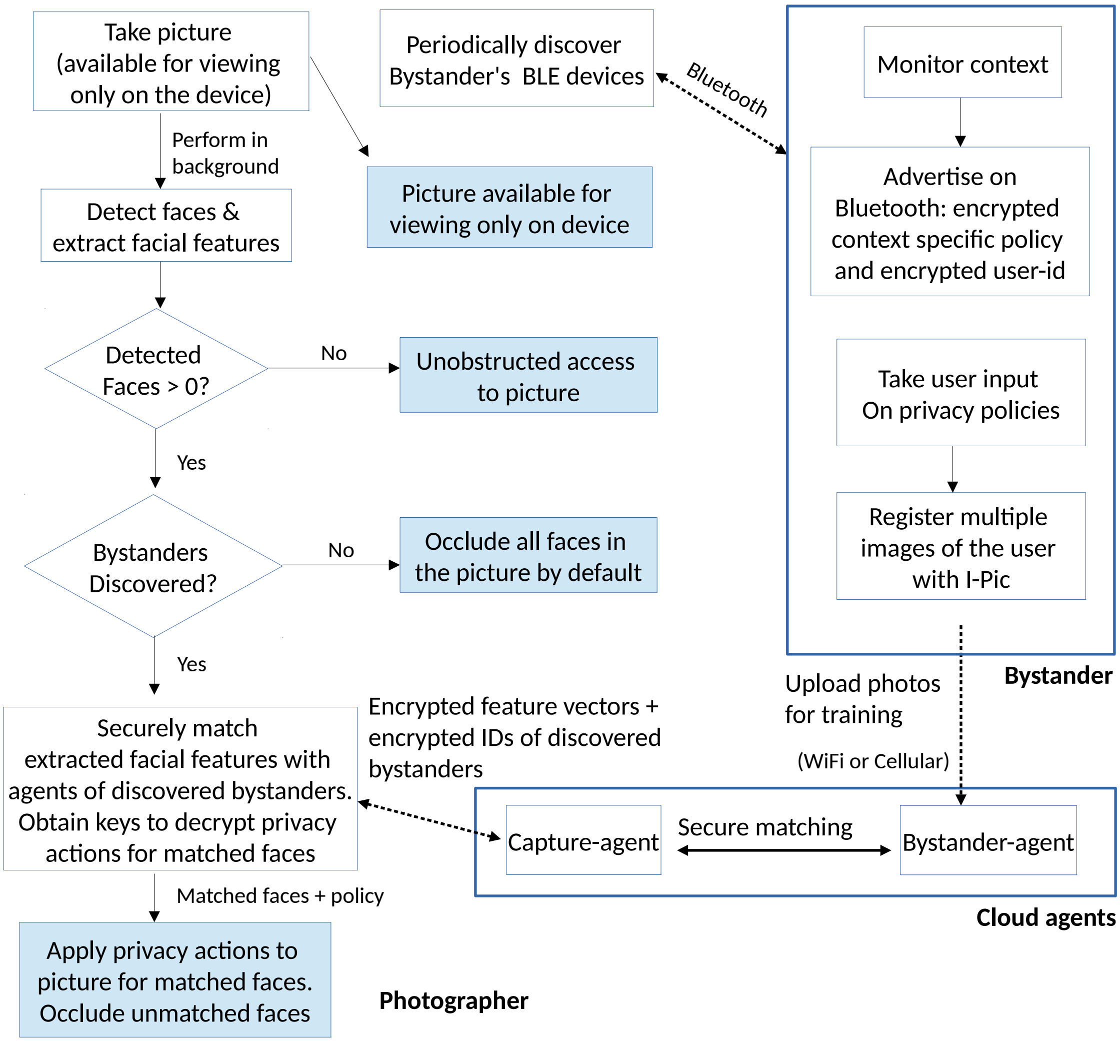

Take picture (available for viewing only on the device)
Perform in background
Detect faces & extract facial features
Detected Faces > 0?
No
Unobstructed access to picture
Yes
Bystanders Discovered?
No
Occlude all faces in the picture by default
Yes
Securely match extracted facial features with agents of discovered bystanders. Obtain keys to decrypt privacy actions for matched faces
Matched faces + policy
Apply privacy actions to picture for matched faces. Occlude unmatched faces
Photographer
Periodically discover Bystander's BLE devices
Picture available for viewing only on device
Bluetooth
Monitor context
Advertise on Bluetooth: encrypted context specific policy and encrypted user-id
Take user input On privacy policies
Register multiple images of the user with I-Pic
Bystander
Upload photos for training
(WiFi or Cellular)
Encrypted feature vectors + encrypted IDs of discovered bystanders
Capture-agent
Secure matching
Bystander-agent
Cloud agents

Figure 6.3: I-Pic workflow

is processed. A background task runs the vision pipeline described below in §6.4.1 to detect faces and extract feature vectors for each. Next, for each feature vector extracted from the image, the background task performs the secure matching protocol described below in *Bystander Agent* §6.4.3 to determine if it matches with the registered classifiers of any of the bystanders in the cache, and decrypts the policies of any matching bystanders.

Finally, the I-Pic background task edits the image according to the policies of the users captured in the image. By default, any face detected in the image that did not match the signature of a bystander is occluded. This conservative choice errs on the side of privacy in case of a bystanders who does not carry a mobile device or does not use I-Pic, whose BLE broadcast was not received, or whose visual signature did not match due to a false negative of the face recognition.

6.4.1 Image processing

The goal of I-Pic's image processing is to identify people captured in the image, extract visual signatures for each person, and match these signatures with those advertised by nearby bystanders.

Face detection. I-Pic must detect faces with high recall, ensuring that bystanders' faces are detected with high probability regardless of size, focus, pose, angle, lighting, or partial occlusion. Unlike the primary subjects of an image, bystanders are not posing for the camera, may be in the background, poorly lit, or out of focus, which makes their detection challenging. We use the open source HeadHunter (Mathias *et al.*, 2014) prototype developed as part of our prior work on face detection. HeadHunter achieves face detection recall of $\sim$95% on standard image datasets like the Annotated Faces in the Wild (AFW, (Zhu and Ramanan, 2012)). For I-Pic, we ported HeadHunter to a mobile tablet with a GPU, as described in §6.5. HeadHunter is superior to other face detectors available for mobile platforms.

Feature extraction. We use the state of the art person recognition method `naeil` from Chapter 3. Unlike typical face recognition systems that can recognise only the frontal faces, our person recognition system has been trained to generalise across head pose by utilizing hairstyle and context information. Since I-Pic aims at identifying bystanders, this person recognition system is highly relevant.

Given a face, the original `naeil` extracts a 4096-dimensional feature vector. To ensure the efficiency of the secure matching algorithm, which is inversely proportional to the number of dimensions, we reduce this feature vector to 128 dimensions by inserting a 128-dimensional fully connected layer before the last layer in the AlexNet, and tune it using the Stochastic Gradient Descent for the person recognition task at hand.

Face recognition. When a user registers, I-Pic extracts `naeil` features from the set of portraits he or she provides. Per-user SVM classifiers are then trained on the features, where positive examples consist of the portraits provided by the corresponding user, and negative examples from the other users and $\sim$12K celebrity faces in the Labelled Faces in the Wild dataset (LFW, (Huang *et al.*, 2007)). On average, there are $\sim$15 positive examples per user, captured with different viewpoints and facial expressions. Users may subsequently provide additional images for training, for instance, if they start to wear glasses or grow a beard.

6.4.2 Cryptographic protocols

I-Pic composes two standard protocols to achieve secure matching: secure dot product and garbled circuits.

Secure dot product. The secure dot product protocol allows two parties, each with a private vector, to compute the vector dot product without divulging the vectors. We use the protocol described in Goethals *et al.* (2004), which is based on the Paillier homomorphic encryption scheme (Paillier, 1999). We use the notation $\llbracket a \rrbracket_{pk}$ to represent the encryption of a number a using a public key pk. The Paillier encryption scheme is additively homomorphic, i.e., given $\llbracket a \rrbracket_{pk}$ and $\llbracket b \rrbracket_{pk}$, it is possible to compute $\llbracket a+b \rrbracket_{pk} = \llbracket a \rrbracket_{pk} \llbracket b \rrbracket_{pk}$. It follows that given $\llbracket a \rrbracket_{pk}$ and an integer c, one can compute $\llbracket ca \rrbracket_{pk} = (\llbracket a \rrbracket_{pk})^c$. These two primitives can be combined to compute the dot product securely. More detail can be found in Goethals *et al.* (2004) and the I-Pic technical report[6].

A straightforward application of this protocol in I-Pic, however, faces two problems: First, the capture device learns the dot products, which would enable a 'rogue' capture device to learn the classifier weight vector of each bystander. By computing dot products using a series of standard basis vectors (vectors that have a value of one in one dimension and zero in all others), the dot product values reveal the dimensions of a bystander's weight vector. To prevent this attack, we use garbled circuits (Yao, 1986), described below, to compute whether the dot product exceeds a threshold $\mathcal{E}$ without revealing the dot product itself.

Second, a capture device typically needs to compare several feature vectors, corresponding to multiple faces that appear in a photo, to the classifier weight vector of a bystander. For n feature vectors with m dimensions, the secure dot product computations require nm encryptions (and n decryptions). We can optimise this computation as follows.

Optimised n x 1 secure dot product. I-Pic reduces the number of encryptions from nm to m using ideas from Huang *et al.* (2011). Consider a matrix V of n vectors with m dimensions each, corresponding to n faces in a photograph, where $V_{i,j}$ is the jth element in the ith vector. Let $c_j = [V_{1,j}, V_{2,j}, ..., V_{n,j}]$ be the jth column of V. The photographer computes an encryption of c_j as $\llbracket c_j \rrbracket_{pk} = \llbracket (V_{1,j}) \,||\, (V_{2,j}) \,||\, ... \,||\, (V_{n,j}) \rrbracket_{pk}$, where || denotes concatenation. This involves only one encryption to produce the ciphertext for n values. The photographer sends $\llbracket \mathbf{c}_1 \rrbracket_{pk}, ..., \llbracket \mathbf{c}_m \rrbracket_{pk}$, the encrypted user ids (*uid*) of the discovered bystanders, and *pk* to the *Bystander Agent*. For each bystander, the *Bystander Agent* computes $\llbracket v_{b_j} \mathbf{c}_j \rrbracket_{pk} = (\llbracket \mathbf{c}_j \rrbracket_{pk})^{v_{b_j}}$ for $1 \leq j \leq m$, where v_b is the classifier weight vector of a bystander. Multiplying these encrypted values, the *Bystander Agent* obtains a packed encryption of the dot products, $\llbracket P_1 \,||\, ... \,||\, P_n \rrbracket_{pk} = \llbracket V_1 \cdot v_b \,||\, V_2 \cdot v_b \,||\, ... \,||\, V_n \cdot v_b \rrbracket_{pk} = \llbracket v_{b_1} \mathbf{c}_1 \rrbracket_{pk} \llbracket v_{b_2} \mathbf{c}_2 \rrbracket_{pk} ... \llbracket v_{b_m} \mathbf{c}_m \rrbracket_{pk}$ and sends it back to the photographer, who decrypts (using *sk*) and unpacks the values to recover the individual dot products.

Garbled circuits for secure threshold computation. Garbled circuits allow two parties holding inputs x and y, respectively, to evaluate an arbitrary function *f(x,y)* without disclosing their inputs. The basic idea is that one party (the garbled circuit generator—the *Capture Agent* in our setting), prepares an "encrypted" version of a

[6] https://people.mpi-sws.org/~paditya/papers/ipic-tr.pdf

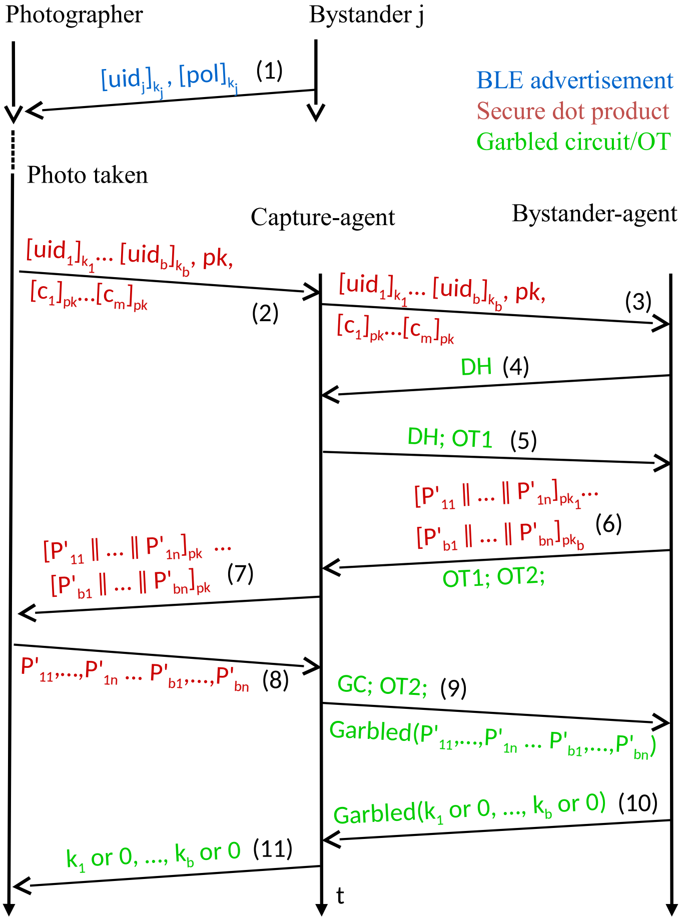

Photographer
Bystander j
$[uid_j]_{k_j}$, $[pol]_{k_j}$ (1)
BLE advertisement
Secure dot product
Garbled circuit/OT
Photo taken
Capture-agent
Bystander-agent
$[uid_1]_{k_1}$... $[uid_b]_{k_b}$, pk,
$[c_1]_{pk}$...$[c_m]_{pk}$
(2)
$[uid_1]_{k_1}$... $[uid_b]_{k_b}$, pk,
$[c_1]_{pk}$...$[c_m]_{pk}$
(3)
DH (4)
DH; OT1 (5)
$[P'_{11} \| ... \| P'_{1n}]_{pk_1}$...
$[P'_{b1} \| ... \| P'_{bn}]_{pk_b}$ (6)
OT1; OT2;
$[P'_{11} \| ... \| P'_{1n}]_{pk}$ ...
$[P'_{b1} \| ... \| P'_{bn}]_{pk}$ (7)
P'_{11},...,P'_{1n} ... P'_{b1},...,P'_{bn} (8)
GC; OT2; (9)
Garbled(P'_{11},...,P'_{1n} ... P'_{b1},...,P'_{bn})
Garbled(k_1 or 0, ..., k_b or 0) (10)
k_1 or 0, ..., k_b or 0 (11)
t

Figure 6.4: I-Pic secure matching protocol for one image with n faces (each facial feature vector has m dimensions). The photographer receives an advertisement from one of b bystanders (blue). The secure dot product computation requires one round trip (red). The garbled circuit (GC) requires a DH key exchange and two rounds of oblivious transfers (OT) (green).

boolean circuit computing f; the second party (the circuit evaluator—the *Bystander Agent* in our case) then obliviously computes the output of the circuit. The combination of *secure dot product* and *garbled circuits* can provide the property that the bystander's session key is revealed to the capture device if, and only if, there is a match between an extracted feature vector and the classifier weight vector of a bystander. The capture device can then decrypt the bystander's policy.

6.4.3 Secure matching protocol

An example message exchange of the secure matching protocol for one image with n detected faces and b bystanders is shown in Figure 6.4. The photographer's device computes the m encrypted column vectors according to the "optimised n x 1" secure dot product protocol, which requires m encryptions. The device sends these vectors to the *Bystander Agent* (via the *Capture Agent*) along with the encrypted user ids of the b bystanders (Message 2 and 3 in Figure 6.4).

The I-Pic *Bystander Agent*[7] now looks up the classifier weight vectors of the b bystanders. For each bystander, it computes the encrypted packed dot products, $[\![P_{i,1} \parallel P_{i,2} \parallel ... \parallel P_{i,n}]\!]_{pk}, 1 \leq i \leq b$, of the bystander feature vector and the n image feature vectors.

The *Bystander Agent* computes *obscured* encrypted packed dot products, $[\![P'_{i,1} \parallel P'_{i,2} \parallel ... \parallel P'_{i,n}]\!], 1 \leq i \leq b$, by adding a different random value $R_{i,j}$ to each dot product $P_{i,j}$, for $1 \leq i \leq b, 1 \leq j \leq n$. This is performed by multiplying each of the b packed encrypted values containing n dot products each, $[\![P_{i,1} \parallel P_{i,2} \parallel ... \parallel P_{i,n}]\!]_{pk}$, with $[\![R_{i,1} \parallel R_{i,2} \parallel ... \parallel R_{i,n}]\!]_{pk}$ for $1 \leq i \leq b$. These *obscured* encrypted packed dot products are sent to the photographer's device via the *Capture Agent* (Message 6 and 7).

The photographer's device decrypts the b packed encrypted values containing n *obscured* dot products each, which requires b decryption operations. The device forwards these obscured dot products to the *Capture Agent* (Message 8), which then constructs a garbled circuit that takes as input n obscured dot products $P'_{i,j} = P_{i,j} + R_{i,j}$, n random values $R_{i,j}$, a session key K_i, and the threshold $\mathcal{E}$ (all provided by the *Bystander Agent*), for $1 \leq i \leq b, 1 \leq j \leq n$. The circuit computes

$$f(P'_{i,j}, \mathcal{E}, R_{i,j}, K_i) = \begin{cases} K_i & \text{if } P'_{i,j} > \mathcal{E} + R_{i,j} \\ 0 & \text{Otherwise} \end{cases}$$

that is, the circuit reveals a bystander's session key if and only if the dot product of the bystander's classifier weight vector and an image feature vector exceed the threshold.

Delivering the *Bystander Agent*'s inputs to the garbled circuit requires a Diffie-Hellman key exchange (DH) and two rounds of oblivious transfers (NPOT, (Naor and Pinkas, 2005) and OTEXT, (Ishai *et al.*, 2003)), which are partly piggy-backed on the secure dot product protocol messages, and shown in Figure 6.4 (Messages 4, 5, 6 and 9). The *Capture Agent* now sends the circuit to the *Bystander Agent*, along with the garbled values of the obfuscated inputs $P'_{i,j}$, and the garbled values of *Bystander Agent*'s inputs as part of the OTEXT oblivious transfer (Message 9). The *Bystander Agent* executes the circuit b times with the appropriate inputs, and returns the garbled results to the *Capture Agent* (Message 10). After ungarbling the results, the *Capture Agent* returns the session keys for the matched bystanders to the photographer's device (Message 11).

As composed, the matching protocol has the desired property that a photographer learns a bystander's current session key if and only if a feature vector in the image matches that bystander's classifier weight vector. Garbled circuits also ensure that the *Bystander Agent* does not learn whether there was a match between the encrypted

[7] To simplify exposition, the description here assumes a single *Bystander Agent* service. The capture device would have to execute the protocol for each *Bystander Agent* in case more than one is discovered.

facial feature vectors and a bystander. Additionally, no principal learns the vectors held by the other principals nor the magnitude of the dot products.

Note that the *Capture Agent* is trusted to construct the garbled circuit correctly. This requirement could be relaxed if one is willing to run additional checks (Lindell, 2013) at some additional computational and runtime overhead.

6.5 EVALUATION

We have prototyped I-Pic on Android version 4.4.2. In our deployment, we used a Google Project Tango Tablet[8] as the photographer's capture device and Galaxy Nexus[9] phones as bystander devices. The Nexus phones advertised their presence once every 640ms over BLE.

We ported HeadHunter (Mathias *et al.*, 2014) to Android for face detection. HeadHunter is optimised for execution on CUDA-enabled GPUs[10]; the Tango Tablet allows us to access CUDA cores. The camera output on the tablet (available as a JPEG file) is first histogram equalised (Lisani *et al.*, 2012) and then resized to 640x360 before being input to HeadHunter. HeadHunter outputs bounding boxes corresponding to detected faces.

To extract feature vectors from facial images, we used an Android port of the Caffe framework[11] and ran it with our FNet neural network. The extracted vectors were normalised such that each feature value was in the range $[0, 1]$. We ported existing Java secure dot product and garbled circuit implementations[12] to C++ on Android to optimise for runtime and energy consumption. The various agents were implemented as HTTP servers.

We begin with a description of I-Pic deployments in various settings; these deployments were also approved by the University of Maryland IRB. While we gained intuition about our vision pipeline using standard face recognition datasets (and the pipeline's performance compares well with the state-of-the-art on them), all results presented here evaluate I-Pic on images captured "in the wild", reflecting spontaneous image capture in different social situations with a range of lighting conditions, camera angles, distances, and poses.

6.5.1 Deployments

To evaluate I-Pic, we registered fifteen volunteers from our institutions using the registration procedure detailed in §6.4.1. Each volunteer received a Galaxy Nexus device for BLE advertisement, which they carried on their person. Registered users

[8] https://store.google.com/?srp=/product/project_tango_tablet_development_kit

[9] Galaxy Nexus has Bluetooth hardware capable of BLE advertising, but the functionality is not available via standard API calls. We patched the kernel to enable BLE advertising.

[10] https://developer.nvidia.com/cuda-zone

[11] https://github.com/sh1r0/caffe-android-lib

[12] http://mightbeevil.org/

Date	Capture device	Number of photographs	Number of ground-truth faces
Nov 20	Tango tablet	81	277
Nov 27	Tango tablet	176	553
Dec 02	DSLR	130	843
	All	387	1673

Table 6.3: Experimental dataset

could choose to either *show* or *blur* their face when photographed; this setting could be changed at their discretion.

The photographs in our results were captured over three days (see Table 6.3), and were taken using the Tango tablet and a DSLR camera. We used the DSLR setup (Sony A7, 35mm f/2.8 lens, 1/80 fixed exposure time with Sony HVL-F32M flash) to simulate better tablet cameras with higher resolution and faster apertures expected in future tablets. The photographs captured by the DSLR were manually fed into the I-Pic processing pipeline.

We annotated all photographs manually with ground truth face rectangles using the open source annotation tool Sloth[13]. For each face, we manually added other information, such as the identity of registered users, pose, and lighting condition.

6.5.2 I-Pic decision tree

In I-Pic, faces in photographs end up being edited (e.g., blurred) or remain unchanged, correctly or incorrectly, depending on decisions made by different subsystems. Figure 6.5 shows the possible paths through I-Pic, culminating in leaf nodes coloured green if I-Pic preserves user privacy and red if it does not. Note that it is possible for I-Pic to make a mistake, e.g., not recognise a face, and for the corresponding path to still lead to a green leaf node, e.g., because the user policy stated not to obscure their face. Finally, some leaf nodes are grey, corresponding to privacy irrelevant mistakes where non-faces were detected as faces and possibly blurred.

Understanding this decision tree, and in particular, analysing where privacy-relevant errors can accrue, will enable us to parametrise and evaluate our vision pipeline in the context of I-Pic's overall goal.

The decision tree has three stages: (1) face detection, (2) face recognition and (3) policy application. Stages 1 and 2 are computational and depend solely on the accuracy of vision pipeline. The diagram separates Stage 3, which is contingent on user choices. For instance, if users choose more permissive policies, then errors from previous stages will less likely result in privacy violations, and vice-versa.

[13] https://cvhci.anthropomatik.kit.edu/~baeuml/projects/a-universal-labeling-tool-for-computer-vision-sloth/

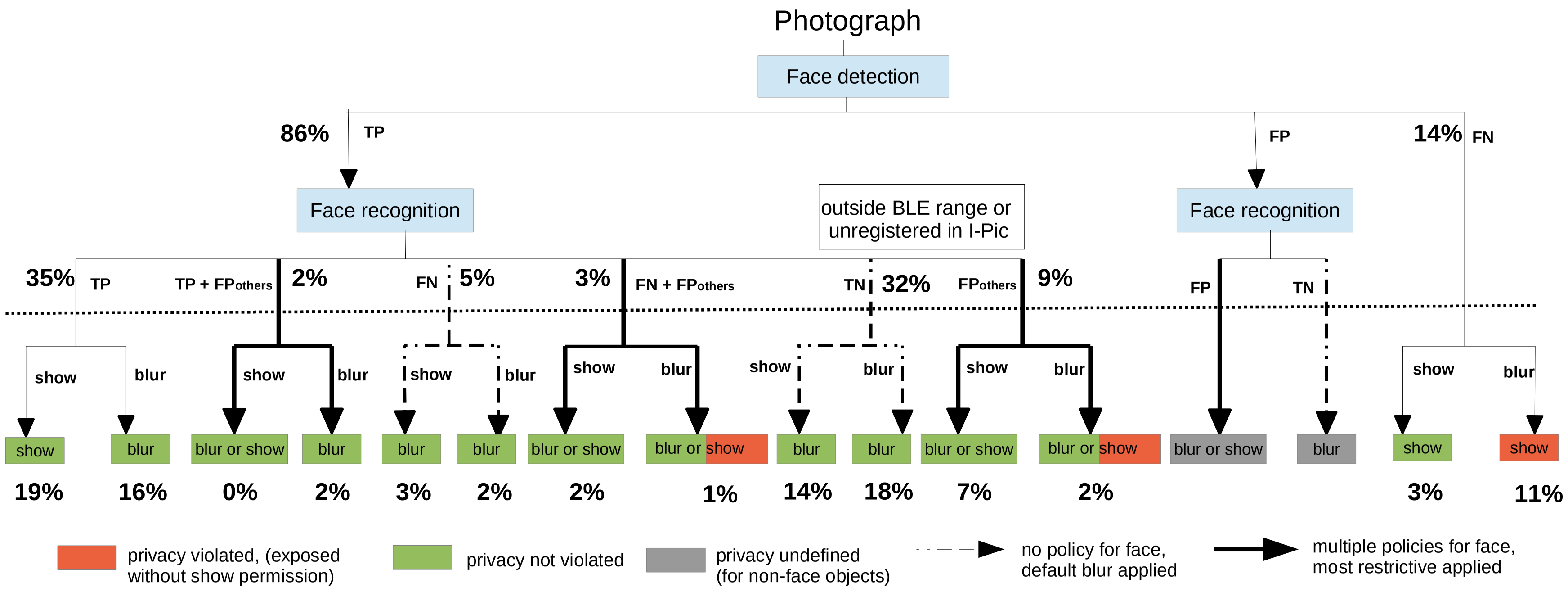


Figure 6.5: I-Pic decision tree

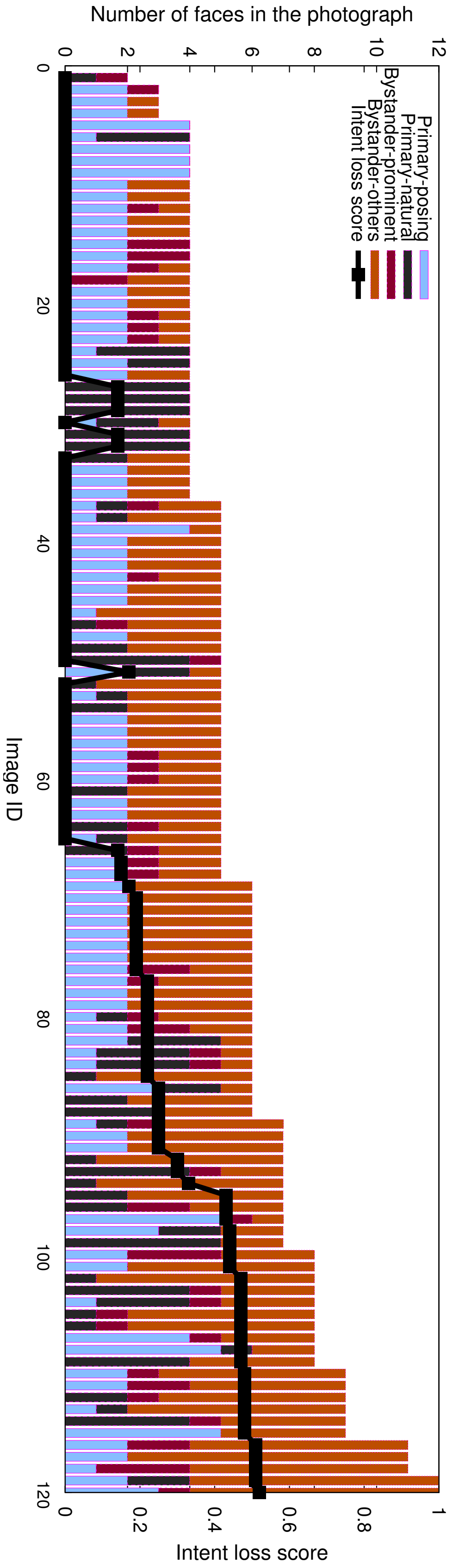


Figure 6.6: I-Pic Intent score of images

Face detection. Stage 1 may result in three outcomes: True Positive (*TP*), where I-Pic detects a face marked in ground truth; False Positive (*FP*), where I-Pic detects a non-face object as a face; or False Negative (*FN*), where I-Pic does not detect a face marked by ground truth. All *TP* and *FP* detections are passed to the face recognition engine in the next stage.

The *FN* faces bypass the I-Pic pipeline and remain unchanged, and can potentially lead to a privacy violation (red leaf node). To minimise these cases, we bias the face detection engine towards higher recall (lower *FN*) at the expense of lower precision (higher *FP*). This means that a non-face object occasionally gets blurred in an image, in exchange for increased privacy.

Face recognition. For a *TP* face detection output, there are six possible choices for recognition in the I-Pic pipeline: (1) True Positive (*TP*), where the detected face is matched only with the individual identified in ground truth; (2) True Positive along with False Positives (*TP**), where the face is matched with the ground truth individual, but also with others[14]; (3) False Negative (*FN*), where the face is not matched with the ground truth person; (4) False Negative along with False Positives (*FN**): I-Pic does not match with the ground truth, but instead matches with one or more other registered individuals; (5) True Negative (*TN*), where I-Pic correctly does not match the face to any registered individual; and (6) False Positive(s) (*FP**), where I-Pic incorrectly matches the face to one or more registered users.

Two leaf nodes have privacy violations for face recognition. *FP* is responsible for both paths, while one of them also requires a *FN*. Thus lower *FP* or high precision has higher priority for recognition, and adequate balance with low *FN* or high recall is also necessary. These requirements guide the parametrisation of the I-Pic face recognition engine.

Misdetected faces (*FP* in detection) are also fed into the recognition protocol, and may lead to (1) True Negatives (*TN*) whereby I-Pic does not recognise the "face" as a registered user, or (2) False Positives (*FP**) where I-Pic mistakenly matches the "face" to one or more registered users.

Policy. Each detected face leads to an action, as shown by the leaves of the tree. If the recognition engine outputs a single user, then the action corresponding to that users' policy is undertaken. However, in cases of multiple matches, e.g., due to *TP**, *FN** or *FP**, the most restrictive policy chosen by any "recognised" user is applied. For all unrecognised users, I-Pic blurs faces by default.

We will detail an experiment with 687 faces in 120 images to examine I-Pic's privacy violations in §6.5.3. The percentages below the leaves in Figure 6.5 show the fraction of faces that mapped to each path in the decision tree, in this experiment. As can be seen from the percentage values, the privacy preferences of 14% of 687 captured faces were violated, primarily due to errors early in the vision pipeline (face detection). In the next sections, we will present detailed evaluations of the

[14] We allow multiple matches; any registered face that exceeds a similarity threshold is considered a match.

Name	Role in photograph	#Occurrences (%)
PP	primary subject posing	185 (26.9%)
PN	primary subject natural	115 (16.7%)
BP	prominent bystander	56 (8.2%)
BO	other bystanders	331 (48.2%)

Table 6.4: Roles of faces captured in images

vision pipeline, whose accuracy primarily determines I-Pic's performance.

6.5.3 I-Pic overall performance

We begin with an evaluation of I-Pic's overall performance in terms of its primary goals, which are to (i) respect bystanders' privacy, and to (ii) preserve the photographer's intent to the extent allowed by subjects' privacy choices.

Toward this end, we took a sample of 120 images with 687 faces marked in the ground-truth. We additionally marked each face according to its role in the image, as shown in Table 6.4, along with the frequency of faces with a given role.

Many of the captured faces correspond to unregistered individuals. Since we don't know the privacy preferences of these individuals, we assigned them policies manually, so that we can process each image as if each captured person were registered with a policy. We assigned the *show-face* policy to the 185 PP faces, since it would be inconsistent for a person who poses for a photograph to refuse to have their face shown. For the remaining 502 faces, we randomly choose one of *show-face* or *blur-face* policies.

The percentage values given at the leaves in Figure 6.5 show what fraction of these 687 faces had what outcome when run through the I-Pic system. As we can see, privacy was violated in 14% of the cases, while the remaining 86% had no privacy violation.

We also assign a privacy loss score in each case of violation. These scores provide a subjective measure of the severity of the privacy violation depending on the role of the face in the image, with higher scores indicating a more severe violation. The privacy loss scores are given in Table 6.5, with the last column indicating how many of each type of violation occurred in the 687 faces.

About 2% of cases had the most severe privacy violation, which is to show a primary subject not posing for the camera against their wishes. Also about 2% of cases had a clearly visible bystander shown against their wishes, and around 10% were less severe cases, where a not prominently depicted bystander was not blurred. We conclude that, overall, I-Pic observes subjects' policies in most cases (86%). Moreover, violations that did occur were mostly in the moderate or mild category.

The second aspect of I-Pic's overall performance is its ability to preserve the photographer's intent, to the extent allowed by the subject's policies. Similar to

Privacy loss score	penalization scenario	#Occurrences
3	PN privacy violated	15 (2.2%)
2	BP privacy violated	12 (1.8%)
1	BO privacy violated	70 (10.2%)
0	no privacy violated	590 (85.9%)

Table 6.5: Privacy loss scores

the privacy loss score, we can define a subjective intent loss score, which penalises blurring a posing primary subject (score 3), blurring a non-posing primary subject with a *show-face* policy (score 2), and bystanders with *show-face* policies (score 1) in decreasing order of severity. The ordering is based on a subjective judgement of intent loss severity when a face is unnecessarily blurred, based on the face's role in the image. We note that our assignment of an intent penalty for the bystander case is conservative, as it is unclear whether a photographer should have expectations about capturing bystanders.

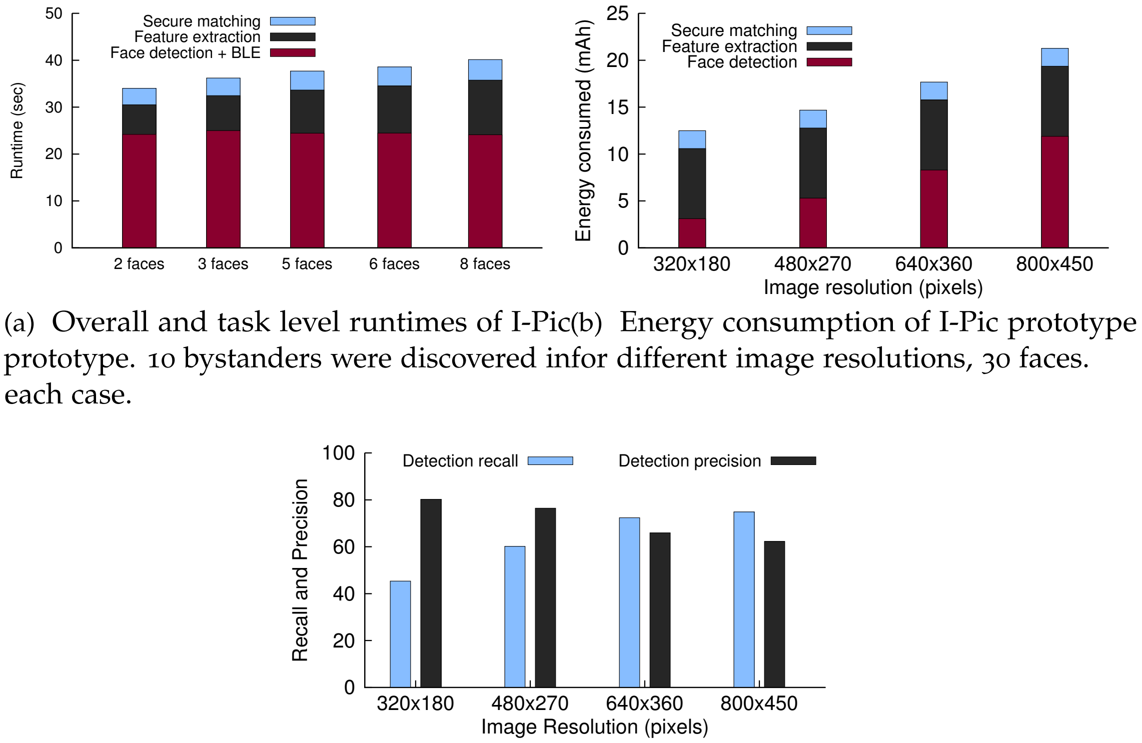
Figure 6.6 shows the intent loss scores for the 120 images, normalised by the maximum intent loss that could occur in a given image. The images are sorted by increasing number of faces from left to right. The bars represent the image composition in terms of roles of the faces depicted in it. I-Pic preserves the photographer's intent, as measured by our score, perfectly in 55 (45.8%) of the images, with the intent loss increasing for pictures with more faces. The vast majority of intent loss cases are caused by a failure to recognise the face of a bystander with a permissive policy, combined with I-Pic's default policy to blur.

Being focused on privacy, I-Pic biases its choices towards privacy, including the default policy and the rule to apply the most restrictive policy in case of multiple matches. As a result, losses in the vision pipeline come at the expense of intent rather than privacy. In the following subsections, we investigate circumstances that lead to imperfections in the vision pipeline, which are causal for the losses in privacy and intent reported here.

6.5.4 Runtime and Energy Consumption

Figure 6.7a plots the overall time taken for I-Pic to process different photographs, along with times spent in different vision and secure matching tasks. In each case, the capture platform received and processed between 3 and 10 BLE advertisements, with varying number of faces in the photograph as plotted along the x-axis. The times for secure matching includes network communication and all cryptographic functions. Face detection dominates, often requiring 25 seconds per photograph. Recall that the processing takes place asynchronously in the background, and does not interfere with the users' experience while capturing and reviewing images.

While the face detection cost in particular is high in our prototype (70–80% of total processing time), we believe it is encouraging that best-of-breed face detection is



(a) Overall and task level runtimes of I-Pic prototype. 10 bystanders were discovered in each case.

(b) Energy consumption of I-Pic prototype for different image resolutions, 30 faces.

(c) Face detection accuracy of I-Pic prototype for different image resolutions.

Figure 6.7: Analysis of performance in terms of computational resources.

feasible on mobile devices available today. Advances in mobile hardware capabilities, driven in part by emerging virtual reality applications, will benefit HeadHunter and other stages of the I-Pic pipeline in the near future. Moreover, face detection is already being offered as a standard feature on mobile platforms, and future implementations (possibly hardware supported) with better accuracy could directly benefit I-Pic.

We measured the energy consumption of the various subcomponents of I-Pic using the Monsoon Power Monitor[15]. We attached the power monitor to a Nvidia Shield Tablet K1[16,17] and processed an image with 30 faces in it. Figure 6.7b shows the energy consumption for different resolutions of the input image. The face detector uses the GPU, whereas the feature extraction is CPU bound. Energy consumption of face detection is independent of the number of faces in an image, whereas it is linear in the number of faces for feature extraction. The secure matching algorithm was run with the 30 faces extracted from the image along with 40 simulated bystanders[18].

Using these measurements, Table 6.6 shows I-Pic's projected capacity on the

[15] https://www.msoon.com/online-store

[16] https://www.nvidia.com/en-us/shield/tablet/

[17] We used the Shield tablet for the power measurements because the Monsoon power monitor is unable to power the Tango tablet. The latter requires a 7.5 volts power supply whereas the Monsoon

Image resolution (pixels)	Number of images processed (containing 30 faces each)
320x180	408
480x270	347
640x360	288
800x450	239

Table 6.6: I-Pic's projected capacity on a 5100 mAh battery

Nvidia Shield tablet, which has a 5100 mAh battery. More than 288 images and 8640 faces can be processed on a single charge. Figure 6.7c compares the face detection accuracy versus the resolution of input images, and serves to highlight the trade-off between accuracy and energy consumption of the prototype. Reducing the resolution to 480x270 pixels enables the prototype to process 20% more images, but comes at a high (12%) drop in face detection recall. On the other hand increasing the resolution to 800x450 only gives diminishing returns for face detection recall when compared to the increased energy consumption that accompanies it.

6.6 conclusion

I-Pic allows users to respect each others' individual and situational privacy preferences, without giving up the spontaneity, ubiquity, and flexibility of digital capture. The I-Pic design and prototype demonstrate that the technical impediments for privacy-compliant imaging can be reasonably overcome using current hardware platforms. I-Pic leverages cutting-edge face detection and recognition technology, which is often perceived as a threat to privacy, to instead increase user's privacy regarding digital capture. Future advances in mobile platform hardware and computer vision will directly benefit I-Pic and further improve the efficiency and accuracy of its I-Pic privacy enforcement.

power monitor can only supply a maximum of 4.5 volts.

[18] BLE scanning for 5 seconds consumes 0.12 mAh of energy, which is accounted for in Figure 6.7b but not shown separately.

Part III

KNOWLEDGE ON TARGET MODEL

For data manipulation to be effective, manipulator's knowledge on the target model is often crucial. This knowledge lets the manipulator focus limited resources (e.g. perturbation size for adversarial examples) on particular aspects of the target. This part discusses ways to represent and increase the manipulator's knowledge on the target model. Discussion and results are relevant to both user privacy and model security.

In Chapter 7 (Oh *et al.*, 2017c), we introduce a Game theoretic framework between two players with opposite goals: the user (manipulator) wants to avoid human identification in her image, while the recogniser wants to re-enable identification. Game theory allows to set a precise level of knowledge on the opponents for each player, and derives the utility guarantees for each player as a function of the knowledge level.

In Chapter 8 (Oh *et al.*, 2018), we seek ways to increase the manipulator's knowledge when the model is a black box (i.e. only query access allowed). We develop a metamodel-based technique, `kennen`, that reverse-engineers certain model hyperparameters only from a set of queries. This in turn narrows down the candidate space for the target, and makes it more vulnerable to e.g. adversarial examples. In user privacy setup, this implies that the manipulator may obfuscate her images more successfully against black-box models, while in security setup, this raises alertness that black-box models are less secure than previously believed.

Chapter 9 (Oh *et al.*, 2017b) is an interlude chapter. Using the fact that activations in an image classifier give reliable cues for the object locations, we train a semantic object segmentation network using only image-level labels. Since the classifier itself cannot give the notion of object shapes, we exploit saliency as the source of the shape prior. In the relevant benchmark, we report the state of the art result among the methods with the same level of supervision.

7 ADVERSARIAL IMAGE PERTURBATION FOR PRIVACY PROTECTION – A GAME THEORY PERSPECTIVE

This chapter presents our final identity obfuscation technique based on adversarial examples, or adversarial image perturbations (AIPs). Recent studies on AIPs suggest that it is possible to confuse recognition systems effectively without unpleasant artifacts. However, AIPs are highly *target specific* – specific knowledge on the target model is required (e.g. gradients) to produce effective perturbations. In practice, the manipulator lacks the knowledge on the target model in general, but even worse, the target model may employ counter measures dynamically against the manipulator (adversarial agent). Game theory provides tools for studying the interaction between agents with uncertainties in the strategies. We introduce a general game theoretical framework for the user-recogniser dynamics, and present a case study that involves current state of the art AIP and person recognition techniques. We derive the optimal strategy for the user that assures an upper bound on the recognition rate independent of the recogniser's counter measure.

We take the user privacy point of view throughout this chapter, but the same analysis also yields the model security guarantees; they are dual problems.

The chapter is based on the paper Oh *et al.* (2017c). As the first author, Seong Joon Oh has conducted all the experiments and has written the conference version manuscript.

7.1 INTRODUCTION

People nowadays share massive amounts of personal photos through social media. Personal photos contain rich private information, e.g. about family members, travel destinations, and political activities. Together with recent developments in computer vision techniques (Deng *et al.*, 2009; Krizhevsky *et al.*, 2012; He *et al.*, 2016; Oh *et al.*, 2015; Sun *et al.*, 2017), this results in increasing concerns that malicious entities employing computer vision technologies could extract private information from visual data.

Classical obfuscation techniques, such as face blurring and pixellisation, is not only unpleasant but also ineffective against convnet-based recognisers (Chapter 4 and Wilber *et al.* (2016); McPherson *et al.* (2016)).

There have been recent studies on *adversarial image perturbations* (AIP): carefully crafted additive perturbations on the image that confuses a convnet while being nearly invisible to human eyes (Szegedy *et al.*, 2014; Goodfellow *et al.*, 2015; Moosavi-Dezfooli *et al.*, 2016, 2017). AIPs are indeed promising as obfuscation techniques.

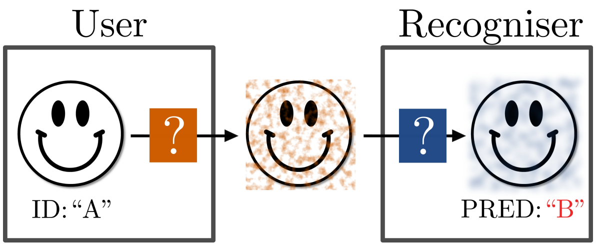


Figure 7.1: A game between a social media user and a recogniser over a photo. The user perturbs the image using orange strategy, trying to confuse the recogniser. The recogniser chooses blue strategy as a counter measure. They do not know which strategy is picked by the other.

However, it remains a question whether AIPs are still effective when counter measures are taken. For example, Graese *et al.* (2016) proposed simple image processing tactics to counter the AIP effects (e.g. blurring by small amount). If furthermore the particular choice of counter measure is unknown, the best strategy is not obvious for the user.

Game theory provides useful tools for analysis when there exist uncertainties in the strategies for each player. We present a game theoretical framework to describe a system in which the user and recogniser strive for antagonistic goals: dis-/enabling recognition. This framework makes it possible to derive guarantees on the user's level of privacy, independent of the recogniser's counter measure, from an explicitly formulated set of assumptions. We include a case study of a person identification game, deriving the user's privacy guarantee with respect to the current state of the art AIP and person recognition methods.

This chapter showcases the utility of game theory in understanding the user-recogniser dynamics. The framework can be extended beyond the particular settings considered. We believe this framework will further aid user-recogniser analyses in more diverse tasks and setups.

The main contributions of the chapter are:

- A game theoretic framework for studying the user-recogniser dynamics.
- Application of *adversarial image perturbation* (AIP) as an effective and aesthetic technique for person obfuscation.
- Novel robust and recogniser-selective AIPs.
- An empirical case study of the game theoretic framework, leading to the privacy guarantees for the user.

7.2 USER-RECOGNISER GAME

This section provides a general framework for studying user-recogniser games. The framework provides a tool for systematising the path from a set of explicit

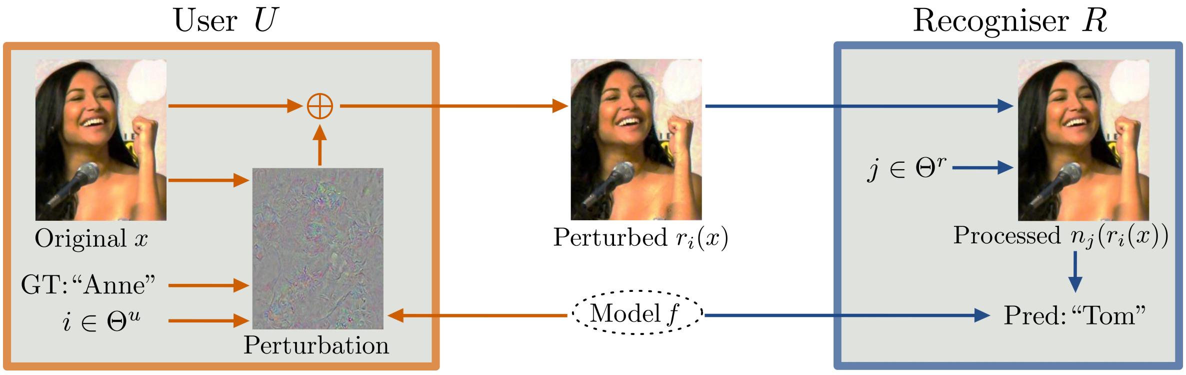


Figure 7.2: User-recogniser game on a single photo. Each player does not know the opponent's strategy. Orange (blue) arrows indicate actions taken by the user (recogniser). Information in the orange (blue) box is only available to the user (recogniser).

assumptions on the players to game theoretical conclusions.

Our user-recogniser game framework is visualised in Figure 7.2. The user U perturbs the original image x according to a strategy $i \in \Theta^u$, aiming to thwart recognition. The recogniser R processes the perturbed image $r_i(x)$ according to a strategy $j \in \Theta^r$, aiming to neutralise the effect of image perturbation. The resulting image $n_j(r_i(x))$ is passed to the model f to make a prediction. The game arises from the fact that each player does not know the opponent's strategy, although they do know each other's strategy space.

We introduce relevant game theoretical concepts and key theoretical results in §7.2.1 to help formalise the framework in §7.2.2. We discuss possible extensions in §7.2.3.

7.2.1 Two-person constant-sum games

We describe our system as a **two-person game** (Neumann, 1928) consisting of two players, the user U and the recogniser R with designated **strategy spaces**, Θ^u and Θ^r. As a result of each player committing to strategies $i \in \Theta^u$ and $j \in \Theta^r$ respectively, R receives a **payoff** of p_{ij}, the recognition rate; U then receives a payoff of $1 - p_{ij}$, the mis-recognition rate. Game theory suggests that it is sometimes better to randomise the strategies. U can adopt a **mixed (random) strategy** $\theta^u = (\theta^u_i)_{i \in \Theta^u}$, defined as a distribution over the strategy space Θ^u, and similarly for R. With abuse of notation we write

$$p(\theta^u, \theta^r) := \sum_{i,j} \theta^u_i \theta^r_j p_{ij} \tag{7.1}$$

for the expected payoff for R when the mixed strategies θ^u and θ^r are taken. The payoff for U is derived and defined as

$$\sum_{i,j} \theta^u_i \theta^r_j (1 - p_{ij}) = 1 - p(\theta^u, \theta^r) =: p'(\theta^u, \theta^r). \tag{7.2}$$

We say that a two-person game is a **constant-sum game** if the players' payoffs sum to a constant β independent of the strategies. In our case, the recognition and mis-recognition rates always sum to one ($\beta = 1$). A game is **finite** if the strategy spaces are finite. We have the following optimality theorem.

Theorem 1 (Neumann (1928))**.** *For a finite constant-sum game, there exist* ***optimal*** *or* ***minimax*** *mixed strategies $\theta^{u\star}$ and $\theta^{r\star}$ such that*

$$p(\theta^{u\star},\theta^{r}) \leq p(\theta^{u\star},\theta^{r\star}) \leq p(\theta^{u},\theta^{r\star}) \quad \forall\, \theta^{u},\theta^{r} \tag{7.3}$$

where $v := p(\theta^{u\star},\theta^{r\star})$ is the ***value of the game****.*

Equation 7.3 implies that when R plays $\theta^{r\star}$, R is guaranteed to have a payoff of at least v, regardless of U's strategy; if U plays $\theta^{u\star}$, U is guaranteed to have a payoff of $1 - v$. In our scenario, this means that U's optimal strategy guarantees a certain mis-recognition rate, regardless of R's strategy.

U's optimal strategies can be obtained efficiently via linear programming that solves the following (R's optimal strategy can be found by swapping min and max):

$$\arg\min_{\theta^u} \max_{\theta^r} \sum_{i,j} \theta^u_i \theta^r_j p_{ij} \quad \text{s.t. } \theta^u,\, \theta^r \text{ are distributions.} \tag{7.4}$$

If U has knowledge on R's strategy $\bar{\theta^r}$, then U can take advantage of this knowledge. U can optimise her strategy given $\bar{\theta^r}$ to attain a payoff of $\max_{\theta^u} p'(\theta^u,\bar{\theta^r}) \geq p'(\theta^{u\star},\bar{\theta^r}) \geq p'(\theta^{u\star},\theta^{r\star}) = 1 - v$, a potentially better payoff than the no-knowledge scenario $1 - v$. However, if R's strategy is optimal $\bar{\theta^r} = \theta^{r\star}$, then the knowledge does not bring improvement for U: $\max_{\theta^u} p'(\theta^u,\theta^{r\star}) = 1 - v$.

In reality, not all players play optimally either due to the lack of knowledge (e.g. on the opponent's strategy space), or due to pure irrationality. We refer to such a player as an **irrational player**. Our discussion above implies:

Corollary 1. *If U knows R's strategy $\bar{\theta^r}$, and if it is suboptimal, then U can enjoy a better payoff than $1 - v$.*

7.2.2 Components of the user-recogniser game

We specify the payoffs, strategy spaces, and information allowed for the user U and the recogniser R.

Test data. We assume that the test data are distributed according to $(\hat{x},\hat{y}) \sim D$. This dataset is the source of private information that the two players compete for.

Fixed model. We assume that U and R use a fixed model f (e.g. a publicly available model). This is a reasonable assumption, as U and R often would not have resources to train modern convnets.

Known model. Each player is aware that the opponent uses f. This may be unrealistic, but provides a good starting point. Relaxation of this assumption is discussed in §7.2.3.

payoff. When the players commit to strategies $i \in \Theta^u$ and $j \in \Theta^r$, R's payoff is the recognition rate on the test set:

$$p_{ij} = \underset{(\hat{x},\hat{y})\sim D}{\mathbb{P}} \left[\arg\max_{y} f^y \left(n_j \left(r_i \left(\hat{x} \right) \right) \right) = \hat{y} \right] \tag{7.5}$$

where f^y denotes the model prediction score for class y. U receives the payoff $1 - p_{ij}$, the mis-recognition rate.

User's strategy space Θ^u. We consider additive perturbations such that for an input x,

$$r_i \left(x \right) = x + t(x), \qquad ||t(x)||_2 \leq \epsilon \tag{7.6}$$

for some constant $\epsilon > 0$. When ϵ is small enough, the perturbation is nearly invisible to human eyes (see Figure 7.3). These perturbations are frequently referred to as *adversarial image perturbations* (AIPs). We discuss existing AIPs and our novel variants in §7.3.

Recogniser's strategy space Θ^r. R aims to neutralise the adversarial effect of AIPs. Although some works have suggested re-training the model with AIPs, demonstrating certain degree of robustification (Goodfellow *et al.*, 2015; Huang *et al.*, 2015), Graese *et al.* (2016) have argued that simple image processing can already neutralise the AIP effects cheaply and effectively. They have demonstrated that on MNIST, translation (T), Gaussian additive noise (N), blurring (B), and cropping & re-sizing (C) have improved the recognition rate from 0% (post-AIP) to 68%, 58%, 65%, and 76%, respectively. In our case study, we will include these transformations in Θ^r. In §7.2.3, we will discuss about expanding strategy spaces.

Known strategy spaces. The strategy spaces for each player (Θ^u and Θ^r) are known to each other, while the chosen strategies are not known.

Multiple recognisers. U may encounter a set of recognisers not all of which are malicious. For example, U uploads her personal photos to a cloud service with a recognition system R_1; she wants an AIP that enables a successful recognition by R_1 but disables recognition by a malicious system R_2. We propose an approach for generating *selective* AIPs in §7.3.2 and confirm their existence in §7.4.5. From a theoretical standpoint, the existence of selective AIPs attest to the diversity of possible AIP patterns, in line with the existence of *universal perturbations* (Moosavi-Dezfooli *et al.*, 2017).

7.2.3 Extensions

In the previous section, we have introduced the user-recogniser game framework with particular assumptions explored in this chapter. In this section, we show that the framework can be extended beyond this setup.

Unknown models. Many AIP techniques assume a full knowledge on the model f, but the computation of *black-box* AIPs is another active research field (Papernot *et al.*, 2016a, 2017; Narodytska and Kasiviswanathan, 2017; Liu *et al.*, 2017a); U can potentially adopt these methods.

Non-constant sum. If U and R assign different weights to different test samples, then the payoffs may not sum to 1. For such non-constant sum games, there exist *Nash equilibrium* strategies for each player (Nash *et al.*, 1950). The optimal strategy and payoff analyses are still possible.

Non-additive AIPs. The framework allows r_i to be any function that induces invisible changes on the image. Current restriction to Equation 7.6 rules out e.g. one-pixel translation of the whole image. Most, if not all, prior work on AIP is done in the additive setup. Crafting non-additive AIP would be interesting future work.

Non-fixed models. R with enough computational resources may re-train the model f with AIPs. One option to expand our framework to such a setup would be to incorporate the model parameters in Θ^r. Brückner *et al.* (2012) have studied this setup, but have assumed convex loss functions. Understanding games with continuous strategy spaces and non-convex payoffs (e.g. convnet losses) is an open question both for computer vision and game theory research.

Unknown strategy spaces. The exact possible set of strategies may not be known to the opponent. With improving technologies, the respective strategy spaces may even grow over time. The framework cannot do much about the unknown strategies, but can adaptively expand the strategy spaces according to technological developments.

7.3 ADVERSARIAL IMAGE PERTURBATION STRATEGIES

This section reviews existing adversarial image perturbation (AIP) algorithms that use first-order optimisation schemes, and proposes our novel variants.

We compute AIPs as additive transformations with L_2 norm constraints (Equation 7.6). Computation of AIP can be formulated as a loss *maximisation* problem

$$\max_t \mathcal{L}\left(f\left(x+t\right), y\right) \qquad \text{s.t. } ||t||_2 \leq \epsilon \tag{7.7}$$

where x is the input image and y is the ground truth label; the loss function $\mathcal{L}$ is to be specified.

Variants	Loss $\mathcal{L}$	Stopping condition	Step size
FGS	$-\log \hat{f}^y$	1 iteration	Fixed
FGV	$-\log \hat{f}^y$	1 iteration	Fixed
BI	$-\log \hat{f}^y$	K iterations	Fixed
GA	$-\log \hat{f}^y$	K iterations	Fixed
DF	$f^{y^c} - f^y$	K it.$\vee$ fooled	Adaptive
GAMAN	$f^{y^\star} - f^y$	K iterations	Fixed

Table 7.1: Conceptual differences among AIP methods. $f^{y'}$ is the model score for class y', and $\hat{f}$ denotes the softmax output of f. y is the ground truth label, and $y^\star$ is the most likely label among wrong ones. y^c is the label with the closest linearised decision boundary.

7.3.1 Existing AIP methods

Depending on the loss function $\mathcal{L}$ and the optimisation algorithm, we recover many of the existing AIP methods such as Fast Gradient Vector (Rozsa *et al.*, 2016), Fast Gradient Sign (Goodfellow *et al.*, 2015), Basic Iterative (Kurakin *et al.*, 2017a), and DeepFool (Moosavi-Dezfooli *et al.*, 2016). The *universal perturbations* introduced by Moosavi-Dezfooli *et al.* (2017) can also be seen as a special case of Equation 7.7 where the loss is computed over the entire test set and the perturbation t is shared across images. See Table 7.1 for the summary.

Fast Gradient Vector (FGV) (Rozsa *et al.*, 2016). FGV adopts the softmax-log loss $\mathcal{L} = -\log \hat{f}^y$ in Equation 7.7, solving it via one-step gradient ascent: $t^\star = -\gamma \nabla \mathcal{L}(x)$ for some constant $\gamma > 0$.

Fast Gradient Sign (FGS) (Goodfellow *et al.*, 2015). FGS is identical to FGV, except that $\nabla \mathcal{L}(x)$ is replaced with $\text{sign}\left(\nabla \mathcal{L}(x)\right)$.

Gradient Ascent (GA). This is a multi-step variant of FGV. Perturbation is initialised at $t^{(0)} = 0$. Gradient ascent is performed on the loss function iteratively: $t^{(m+1)} = t^{(m)} - \gamma \nabla \mathcal{L}(x + t^{(m)})$ for $m = 0, \cdots, K$ for some fixed step size $\gamma > 0$ and maximal number of iterations $K \geq 1$.

Basic Iterative (BI) (Kurakin *et al.*, 2017a). BI is identical to GA, except that $\nabla \mathcal{L}(x)$ is replaced with $\text{sign}\left(\nabla \mathcal{L}(x)\right)$.

DeepFool (DF) (Moosavi-Dezfooli *et al.*, 2016). DF algorithm solves the objective:

$$\min_t ||t||_2 \quad \text{s.t.} \quad \arg\max_y f^y(x+t) \neq y \tag{7.8}$$

which finds the minimal perturbation such that the prediction is wrong. Although the objective is different, we show that the DF algorithm can also be seen as a first-order method solving Equation 7.7 for some loss function.

DF first finds the class with the nearest decision hyperplane, denoted by c. To simplify the search, c is found on the linear approximation of f around x (tangent function). The normal vector to the decision hyperplane is given by $\nabla f^c - \nabla f^y$. At each iteration, the algorithm computes the minimal step size along this direction to reach the decision hyperplane. Since f is not linear, the algorithm may need more than one iterations to cross the decision hyperplane.

We observe that if we set the loss function as $\mathcal{L} = f^c - f^y$ the gradient ascent direction matches the DF step directions $\nabla f^c - \nabla f^y$. We thus regard DF as a gradient ascent algorithm with each step size minimised to just induce a wrong prediction.

Projection and clipping. The norm constraint $||\cdot||_2 \leq \epsilon$ as well as RGB value constraint to $[0, 255]$ must be enforced on the solution. Liu *et al.* (2017a); Kurakin *et al.* (2017a) suggest applying projections after each iteration. We follow this practice. For BW images, we average the gradients for each RGB channel.

7.3.2 Our AIP methods

As we will demonstrate in §7.4.2, the above approaches are fragile to simple image processing techniques. We propose novel AIP approaches here, focusing on robustness.

Gradient Ascent – Maximal Among Non-GT (GAMAN). Even if the prediction label is changed by the AIP, this would not be robust if the perturbed input is still close to the decision boundary. DeepFool (DF) is not expected to be robust, as it stops iterations as soon as the decision boundary is reached. On the other hand, DF guides the solution to the closest decision boundary; if we let DF iterate beyond the decision boundary with a fixed step size with fixed number of iterations, the solution is likely to proceed more deeply into the territory of the wrong label, improving robustness.

This motivates our GAMAN[19] variant. Instead of the costly computation of c at each iteration, we approximate

$$c \approx y^\star := \arg\min_{y' \neq y} f^{y'} \tag{7.9}$$

the most likely prediction among wrong labels. We set the loss function as $\mathcal{L} = f^{y^\star} - f^y$, and perform gradient ascent with a fixed step size γ for K iterations. This approach is similar but different from the impersonation AIPs previously considered (Sharif *et al.*, 2016; Liu *et al.*, 2017a), which drive the solution to a fixed impersonation target $\bar{y}$. In contrast, $y^\star$ may change during the iterations.

Vaccination against image processing. The above methods maximise classification loss functions with respect to a fixed recogniser. For countering an AIP-neutralising

[19] *Gaman* is a Zen Buddhist term for *endurance*.

image processing technique n_j, we consider including the image processing step in the loss function: $\mathcal{L}(n_j(x+t))$. Any first-order method considered above can be used, as long as n_j is differentiable. If the processing function is random, we average the gradients from multiple samples. We refer to this technique as *vaccination*. Note that this technique is complimentary to the above mentioned methods.

Selective AIPs. We present another complimentary technique for generating AIPs targeted to a selected subset of recognisers. To avoid recognition from $\mathcal{M}$ while authorising $\mathcal{B}$ to recognise, we propose to maximise a mixed loss

$$\sum_{k\in\mathcal{M}} \lambda_k \mathcal{L}_k - \sum_{k'\in\mathcal{B}} \lambda_{k'} \mathcal{L}_{k'} \tag{7.10}$$

with $\lambda_k, \lambda_{k'} > 0$.

7.4 EMPIRICAL STUDIES

We have set up a game theoretical framework to study the dynamics between the user U and the recogniser R. In particular, previous adversarial image perturbation (AIP) techniques are studied, and new variants are proposed.

In this section, we present a case study of the framework on *person recognition*. Before presenting the game theoretical analysis, we evaluate the performance of existing and newly proposed AIP techniques (§7.4.2), and the effectiveness of R's image processing strategies Θ^r (§7.4.3). The full game is introduced (§7.4.4) after specifying U's strategy space; we study this system in depth. Finally, we show results on the recogniser-selective AIPs (§7.4.5).

7.4.1 Dataset and Experimental Setup

Dataset. We build our analysis upon the PIPA dataset (Chapter 3 and Zhang *et al.* (2015b)). A large-scale dataset of Flickr personal photos, PIPA provides a realistic testbed for identity obfuscation in the social media setup. We use the $val_{0/1}$ Original split (see Chapter 3), consisting of 4820 instances of 366 identities as the training and test sets. We assume that the user uploads cropped head images to social media; we use the head bounding box annotations in PIPA.

Person recogniser. The person recognition model f is built on the person recogniser `naeil` (Chapter 3). Unlike the original version that is built on AlexNet (Krizhevsky *et al.*, 2012), we also consider VGG (Simonyan and Zisserman, 2015), GoogleNet (Szegedy *et al.*, 2015), and ResNet152 (He *et al.*, 2016) as feature extractors. They show better recognition rates (Table 7.2).

Evaluation. We evaluate payoffs for R in terms of the ratio of correctly identified instances in the test set. The payoff for U is 1 minus R's payoff. In all the tables, R is the column player and U is the row player. For each column (row), U's (R's) optimal strategy is marked orange (blue).

	Perturbation	AlexNet	VGG	Google	ResNet
	None	83.8	86.1	87.8	91.1
Image Proc.	Noise	≥83	≥85	≥87	≥90
	Blur	≥82	≥85	≥86	≥90
	Eye Bar	≥81	≥84	≥84	≥87
1-Iter. AIP	FGS	23.6	16.0	5.9	20.2
	FGV	13.3	11.5	4.6	20.0
K-Iter. AIP	BI	1.2	0.5	0.0	0.0
	GA	0.2	0.0	0.0	0.0
	DF	0.0	0.0	0.0	0.0
	GAMAN	0.0	0.0	0.0	0.0

Table 7.2: Recognition rates after image perturbation. In all methods, the perturbation is restricted to $||\cdot||_2 \leq 1000$. For the baseline image processing perturbations, we only report lower bounds (denoted $\geq \cdot$).

7.4.2 Comparison of perturbation methods

AIP parameters. We set $\epsilon = 1000$ in all our experiments, unless stated otherwise. For GoogleNet input 224×224, this corresponds to 2% of pixels perturbed by 1/256. For Gradient Ascent (GA) and Basic Iterative (BI) the step size γ is set to 10^4; for GAMAN, 5×10^3. We set the maximal number of iterations $K = 100$, determined such that the norm reaches $\epsilon = 1000$ in K iterations for most test samples.

Baseline perturbation methods. We consider three commonly used obfuscation types: noise, blur, and eye bar. Noise adds iid Gaussian noise of variance σ^n; blur performs convolution with a Gaussian kernel of size σ^b; eye bar puts a gray horizontal bar of thickness σ^e on the upper $\frac{1}{3}$ location. They incur large L_2 distances (>1000) from the original image even with small σ^n, σ^b, and σ^e. In Table 7.2, we report the *lower bounds* on the recognition rates at $||\cdot||_2 = 1000$ by computing the rates at some $||\cdot||_2 > 1000$.

AIP performance. We first evaluate all the considered AIP methods against all network variants. Table 7.2 shows the results. We observe that noise, blur, and eye bar have nearly no impact on the recognition performance for small L_2 perturbations. AIP variants show better obfuscation performances. Vanilla gradient overall gives better obfuscation than signed versions; on AlexNet Fast Gradient Vector (FGV) reduces the recognition rate to 13.3, compared to 23.6 for Fast Gradient Sign (FGS); the multi-iteration analogues show similar behaviours with Gradient Ascent (GA) achieving 0.2 compared to 1.2 by Basic Iterative (BI). Finally, we observe that the DeepFool (DF) and GAMAN (§7.3.2) are very effective, pushing the recognition rates down to zero.

Perturbation	∅	Proc	T	N	B	C	TNBC
None	87.8	87.8	87.6	64.0	81.2	85.4	87.3
BI	0.0	8.3	15.8	16.8	28.6	27.4	17.6
GA	0.0	8.6	13.2	14.1	28.4	23.7	16.4
DF	0.0	51.8	75.6	56.5	72.5	76.9	75.5
GAMAN	0.0	4.0	6.6	15.0	22.2	16.7	9.9

Table 7.3: Robustness analysis of AIPs on GoogleNet. AIPs are restricted to to $||\cdot||_2 \leq 1000$. Proc indicates the re-sizing and quantisation needed to convert AIP outputs to image files. $(\mathsf{T},\mathsf{N},\mathsf{B},\mathsf{C})$ = (Translate, Noise, Blur, Crop).

Network performance. Comparing architectures, we observe that AlexNet is surprisingly robust to AIPs compared to more recent architectures. GoogleNet, for example, performs better than Alexnet without AIPs (83.8 vs 87.8); when FGS is used, AlexNet performs 23.6 while GoogleNet performs 5.9. When multi-iteration AIPs are used, the architectural choice does not have a significant impact. We opt for GoogleNet in the next experiments; it is reasonably performant, while being much faster than ResNet.

7.4.3 Robustness of AIPs

Basic processing Proc**.** Even before R's image processing strategies take place, the perturbed image needs to be (1) re-sized to the original image (from the network input sizes) and (2) quantised to integer values (e.g. 24-bit true colour). We denote the above two basic processing steps as Proc.

Image processing strategies Θ^r. We fully specify R's strategy space for our case study. Following Graese *et al.* (2016), we consider $\Theta^r = \{\mathsf{Proc}, \mathsf{T}, \mathsf{N}, \mathsf{B}, \mathsf{C}, \mathsf{TNBC}\}$. Proc is the basic processing described above, and all the other strategies are applied over Proc. T is translation by a random offset within 10% of the image side lengths. N adds iid Gaussian noise with variance $\sigma^2 = 10^2$. B blurs with Gaussian kernel of width chosen from $\{1,3,5,7,9\}$ uniformly at random. C crops with a random offset within 10% of the image side lengths and re-sizes back to the original. For each strategy, the recogniser ensembles the scores from five random samples. We also consider the combination of all four (TNBC). It runs the model four times on each processed image and once on the original; the scores are then averaged.

Robustness of AIPs. Table 7.3 shows the recognition rates for the GoogleNet when R's processing strategies are present. While the multi-iteration AIPs induce zero recognition rates without any processing, Proc already exhibits powerful neutralisation effects: recognition rates for Gradient Ascent (GA) and DeepFool (DF) jump from zero to 8.6 and 51.8, respectively. The instability of DF is due to early stopping (§7.3.1). The processing strategies by R further increase recognition rates. Blurring B and

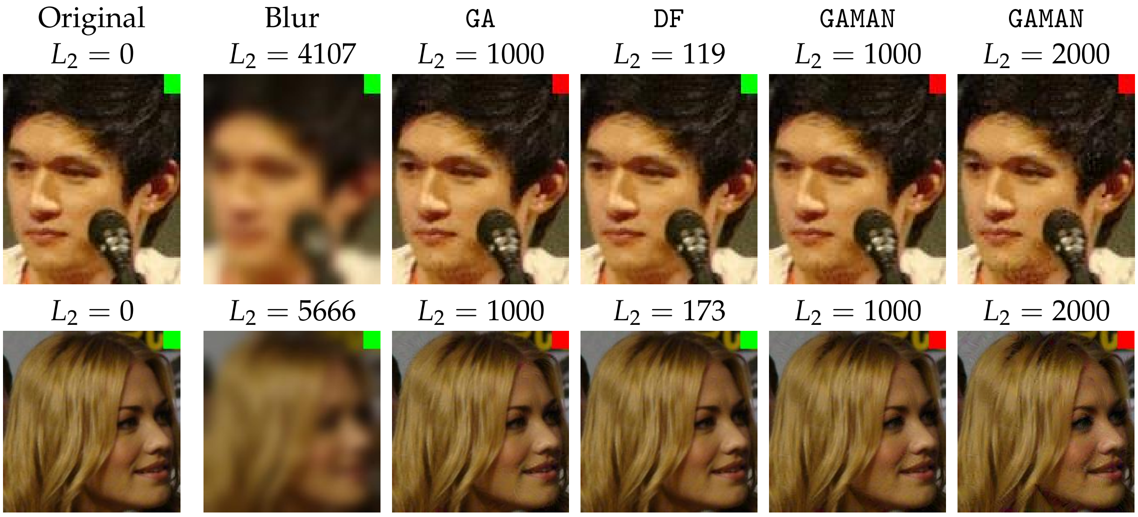


Figure 7.3: Perturbed images after Proc and the corresponding predictions (green for correct, red for wrong). GA and GAMAN reliably confuse the classifier at almost no cost on the aesthetics. At $L_2 = 2000$, GAMAN does show small artifacts.

cropping C strategies prove to be more harmful to AIPs than translation T and noise N in general. Comparing AIP-wise, we show that our novel variant GAMAN (§7.3.2) dominates other methods against all processing strategies but N; GA performs better in that case, but only by a small amount (14.1 versus 15.0). Subsequent analyses are built on GAMAN.

Qualitative. Qualitative examples of the methods are shown in Figure 7.3. The images and the prediction results are after Proc. GA and GAMAN reliably induces misidentification without sacrificing aesthetics compared to blurring.

AIP Performance at Different L_2 Norms. We examine the behaviour of fooling effects with respect to the L_2 norm size ϵ. See Figure 7.4 for the plot. The performances are post-Proc (§7.4.3). We fix the step size to $\gamma = 10^4$ (5×10^3 for GAMAN), and the maximal number of iterations to $K = 100$; we choose the norm constraint ϵ from $\{100, 200, 500, 1000, 2000\}$. The norm of the resulting AIP is upper bounded by ϵ, but may not necessarily be exactly ϵ. The average norm across the test set is plotted.

We observe that the AIP variants are much more effective than Noise, Blur, or Eye Bar, achieving the same degree of obfuscation at $1 \sim 2$ orders of magnitude smaller perturbations. At the same norm level, the multi-iteration variants (BI,GA) are more effective than the single-iteration analogues (FGS,FGV). Taking gradient signs decreases the obfuscation performance at small L_2 norms (≤ 1000), but they converge to a similar performance at $\epsilon = 2000$. DeepFool (DF) outputs have small norms ≤ 100 due to early stopping. Our variant GAMAN performs best across all norm levels, achieving nearly zero recognition at $\epsilon = 2000$.

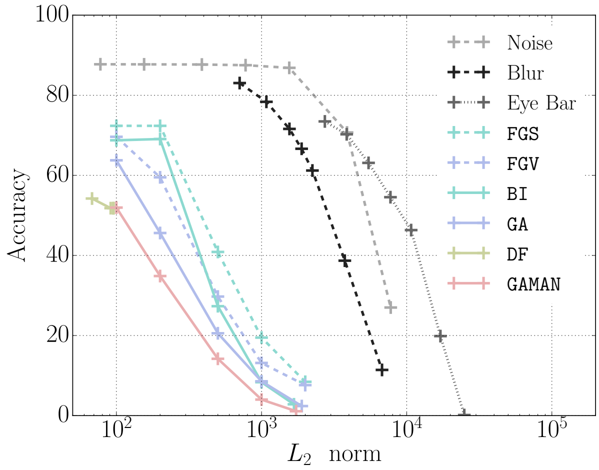


Figure 7.4: GoogleNet accuracy after various perturbations methods at different L_2 norms. All results are after Proc.

7.4.4 User-recogniser games

Vaccination strategies Θ^u. In response to the processing strategies by the recogniser R, the user U may vaccinate the AIP against expected processing types (§7.3.2). We consider six variants $\Theta^u = \{\texttt{GAMAN}, \texttt{/T}, \texttt{/N}, \texttt{/B}, \texttt{/C}, \texttt{/TNBC}\}$. We use slash / to indicate vaccination on GAMAN. For /T, /N, /B, /C, gradients from 5 random function samples are averaged at each iteration. The combination strategy /TNBC averages 4 gradients from individual methods and 1 original gradient, resulting in the same number of gradient computations for all vaccination variants.

Is vaccination helpful?. Table 7.4 shows the recognition rates of GoogleNet for combinations of discussed processing and vaccination strategies. We observe indeed that each vaccination type makes the vanilla AIP GAMAN more robust against the respective processing type: for B the rate drops from 22.2 to 5.8. /B is the most effective strategy for U against all processing strategies except for N. For N, the corresponding vaccination /N yields the best payoff for U. We conjecture this is because the noise N results in high frequency patterns while the others smooth the output. We observe, finally, that the combined vaccination /TNBC cannot prepare AIP against all processing types most effectively; given a budget on the number of gradient computations, it is hard to be good at everything.

Optimal deterministic strategy. We can regard Table 7.4 as the payoff table p_{ij} for R for strategies $i \in \Theta^u$ and $j \in \Theta^r$. Let's first assume that the players only choose fixed strategies. Then, solving Equation 7.4 with determinism constraints $\theta^u_i, \theta^r_j \in \{0,1\}$ yields U's optimal strategy as /B with a privacy guarantee of at most 8.6 recognition rate.

	Recogniser Θ^r					
User Θ^u	Proc	T	N	B	C	TNBC
`GAMAN`	4.0	6.6	15.0	22.2	16.7	9.9
`/T`	2.5	2.3	11.6	18.5	7.2	4.9
`/N`	5.8	7.6	4.6	23.6	16.6	9.1
`/B`	0.4	0.8	8.6	5.8	3.1	1.4
`/C`	2.6	2.2	11.8	18.1	3.4	4.3
`/TNBC`	0.7	0.9	5.2	9.5	3.2	2.0

Table 7.4: Recogniser's payoff table p_{ij}, $i \in \Theta^u$ and $j \in \Theta^r$. The user's payoff is given by $100 - p_{ij}$.

Optimal random strategy. Game theory suggests that it is sometimes better to randomise strategies. Solving Equation 7.4 without the integral constraints yield the optimal solutions for U and R as $\theta^{u\star} = (\texttt{/B} : 61\%, \texttt{/TNBC} : 39\%)$ and $\theta^{r\star} = (\mathsf{N} : 52\%, \mathsf{B} : 48\%)$, respectively. Playing $\theta^{u\star}$ guarantees U to allow at most 7.3 recognition rate, an improved privacy guarantee than the deterministic case, 8.6.

Knowledge on *R*'s strategy. As discussed in §7.2.1, having knowledge on R's strategy can improve the payoff bound for U, if R does not play the optimal strategy. Let us consider two possible non-optimal strategies played by R. (1) If R commits to B, U's optimal strategy is the minimal row in the column B: `/B`, with recognition rate 5.8. (2) If R randomises uniformly over Θ^r, U's optimal strategy is the minimal row over the column average: `/B` with recognition rate 3.4. In both cases, U enjoys lower recognition rates.

Limited knowledge on Θ^r. Assume that U is not aware of all possible technologies that R has at hand. For example, the strategy N is not known to U. Then, U's apparent optimal solution is $(\texttt{/B} : 100\%)$, which she thinks will guarantee her at most 5.8 recognition rate. R can then attack U with N, incurring 8.6 recognition rate. Limited knowledge on the opponent's strategy space does hurt.

7.4.5 Selective AIPs

We assume that U wants to avoid identification by a set of malicious recognisers $\mathcal{M}$, while authorising identification by benign ones $\mathcal{B}$. We set up the experiments in Table 7.5. We include the `GAMAN` performance on GoogleNet as a baseline (first row). We solve Equation 7.10 with $\lambda_k = 1$ for all $k \in \mathcal{M} \cup \mathcal{B}$ to generate selective AIPs.

When $\mathcal{M} = \{\text{GoogleNet}\}$ and $\mathcal{B} = \{\text{AlexNet}\}$, the generated AIP incurs mere 8.7 identification for $\mathcal{M}$ (after Proc), while allowing $\mathcal{B}$ to identify 97.9 percent. We thus confirm the selectivity. However, this comes at the cost of increased recognition rate for $\mathcal{M}$ (8.7), compared to when AIP only had to confuse $\mathcal{M}$ (4.0).

We also consider the multi-$\mathcal{M}$, multi-$\mathcal{B}$ case given by $\mathcal{M} = \{\text{AlexNet, ResNet}\}$

Setup			$\mathcal{M}$ averaged		$\mathcal{B}$ averaged	
$\mathcal{M}$	$\mathcal{B}$	L_2	w/o AIP	w/ AIP	w/o AIP	w/ AIP
{G}	$\emptyset$	1000	87.8	4.0	-	-
{G}	{A}	1000	87.8	8.7	83.8	97.9
{A,R}	{V,G}	1000	87.4	17.7	87.0	97.7
{A,R}	{V,G}	2000	87.4	3.8	87.0	97.8

Table 7.5: Selective AIPs. AIPs are crafted to confuse $\mathcal{M}$ leaving $\mathcal{B}$ intact. [A,V,G,R] = [AlexNet, VGG, GoogleNet, ResNet152]. GAMAN has been used in all experiments. Reported performances are after Proc.

and $\mathcal{B} = \{\text{VGG, GoogleNet}\}$. The average performance is 17.7 for $\mathcal{M}$, and 97.7 for $\mathcal{B}$, post Proc. Selectivity thus works for multiple models, but again the recognition rates for $\mathcal{M}$ are quite high (17.7). We remark that by increasing the budget on perturbation size from 1000 to 2000, we can still attain a lower rate: 3.8.

The existence of selective AIPs is not only of practical but also of theoretical interest. They show that the space of AIPs is diverse enough to accommodate patterns that simultaneously hamper and assist recognition.

7.5 DISCUSSION & CONCLUSION

Game theoretical approach. Game theory is a tool for wading through uncertainties in players' choices, providing payoff guarantees independent of the opponent's strategies. Game theory also suggests that if there is no single technology which best copes with all possible adversarial technologies, it is better to randomise existing techniques.

As discussed in §7.2.3, the game theoretical framework introduced in this work can be extended to other setups, where less resource constraints are placed on each player. This work serves as a first step towards the promising research direction of analysing the user-recogniser dynamics.

Conclusion. In this chapter, we have constructed a game theoretical framework to represent the manipulator's uncertainty on the target model. Game theoretical analysis yields privacy bounds for the user privacy. We note that the model security problem is a dual problem of the user privacy problem and that we obtain the model security bound on the way as well.

7.6 ADDITIONAL RESULTS

In §7.4, we have focused on the GoogleNet results for the AIP robustness analysis and the game theoretic studies (Tables 7.3 and 7.4). We show extended results on AlexNet, VGG, and ResNet152. We also show more qualitative examples, extending

Network	Optimal Strategy $\theta^{u\star}$	Bound on Rec. Rate
AlexNet	$($/B$: 100\%)$	$\leq$6.4
VGG	$($/B$: 86\%,$ /TNBC$: 14\%)$	$\leq$4.9
GoogleNet	$($/B$: 61\%,$ /TNBC$: 39\%)$	$\leq$7.3
ResNet	$($/B$: 31\%,$ /TNBC$: 69\%)$	$\leq$8.5

Table 7.6: Optimal strategies and the corresponding guaranteed upper bounds on the recognition rate for different networks. We write $\leq \cdot$ to denote the upper bound.

Figure 7.3.

7.6.1 Robustness analysis

See Table 7.7a for the robustness analyses for the three other networks. We confirm here again that GAMAN shows overall best robustness, across image processing techniques (Proc, T, N, B, C, and TNBC), across architectures. For AlexNet and ResNet, cropping (C) is the most powerful neutralisation, while for VGG and GoogleNet blurring (B) is. We observe that the effects are particularly strong for ResNet; C boosts the performance from 0.0 to 31.8 against GAMAN.

7.6.2 Game analysis for various networks

See Table 7.7b for the payoff tables for the three other networks. We summarise the optimal user strategy $\theta^{u\star}$ and the corresponding guarantee on the recognition rate in Table 7.6. Note that against all but AlexNet architecture, the optimal strategy $\theta^{u\star}$ is given as a mixture of /B and /TNBC.

7.6.3 More qualitative results

We include more qualitative results (extension of Figure 7.3). See Figures 7.5, 7.6, and 7.7. GA and GAMAN reliably confuse the classifier at almost no cost on the aesthetics. As the L_2 norm increases, artifacts become more visible.

AlexNet

Perturb	∅	Proc	T	N	B	C	TNBC
None	83.8	83.8	83.7	77.8	78.7	80.1	83.9
BI	1.2	10.0	29.7	20.8	26.6	34.3	23.3
GA	0.2	4.8	13.6	11.6	17.7	17.8	12.2
DF	0.0	62.1	76.5	68.5	69.4	75.0	74.7
GAMAN	0.0	1.4	6.4	9.2	13.5	12.3	5.6

VGG

Perturb	∅	Proc	T	N	B	C	TNBC
None	86.1	86.1	84.8	77.2	81.5	84.1	85.8
BI	0.5	6.8	11.1	18.1	23.2	16.8	14.4
GA	0.0	4.2	5.5	11.2	17.2	10.2	8.2
DF	0.0	53.3	66.3	65.9	69.4	69.2	71.4
GAMAN	0.0	1.6	2.1	8.5	11.8	5.6	3.5

ResNet

Perturb	∅	Proc	T	N	B	C	TNBC
None	91.1	91.1	90.6	72.0	87.2	89.3	90.8
BI	0.0	10.9	36.8	24.8	32.8	45.3	26.3
GA	0.0	15.2	37.3	24.4	36.9	43.7	28.9
DF	0.0	52.9	83.1	65.0	76.8	84.2	80.9
GAMAN	0.0	7.3	23.4	23.3	28.2	31.8	18.4

(a) Robustness analysis for AIPs.

AlexNet

	Recogniser Θ^r					
User Θ^u	Proc	T	N	B	C	TNBC
GAMAN	1.4	6.4	9.2	13.5	12.3	5.6
/T	0.9	0.8	6.2	10.5	2.7	2.2
/N	1.2	4.2	4.8	11.7	9.5	3.9
/B	0.8	3.5	6.3	6.4	6.0	2.6
/C	2.4	2.5	9.2	13.1	1.3	3.4
/TNBC	0.6	1.2	4.5	7.8	2.9	1.9

VGG

	Recogniser Θ^r					
User Θ^u	Proc	T	N	B	C	TNBC
GAMAN	1.6	2.1	8.5	11.8	5.6	3.5
/T	1.5	1.2	8.1	12.3	3.2	2.8
/N	2.0	2.5	3.9	12.6	6.7	3.9
/B	0.3	0.7	5.0	4.5	2.2	1.2
/C	2.0	1.6	9.5	14.0	1.9	3.1
/TNBC	0.6	0.7	4.3	7.3	2.3	1.4

ResNet

	Recogniser Θ^r					
User Θ^u	Proc	T	N	B	C	TNBC
GAMAN	7.3	23.4	23.3	28.2	31.8	18.4
/T	2.9	2.8	16.6	19.0	5.4	5.8
/N	5.3	12.9	4.2	23.5	20.1	10.2
/B	0.6	3.1	13.0	6.8	5.3	2.4
/C	3.5	3.1	17.0	18.8	3.2	5.4
/TNBC	0.7	1.2	6.5	9.3	2.9	2.3

(b) Payoff tables.

Table 7.7: Extended version of Table 7.3 and 7.4 for the other network architectures.

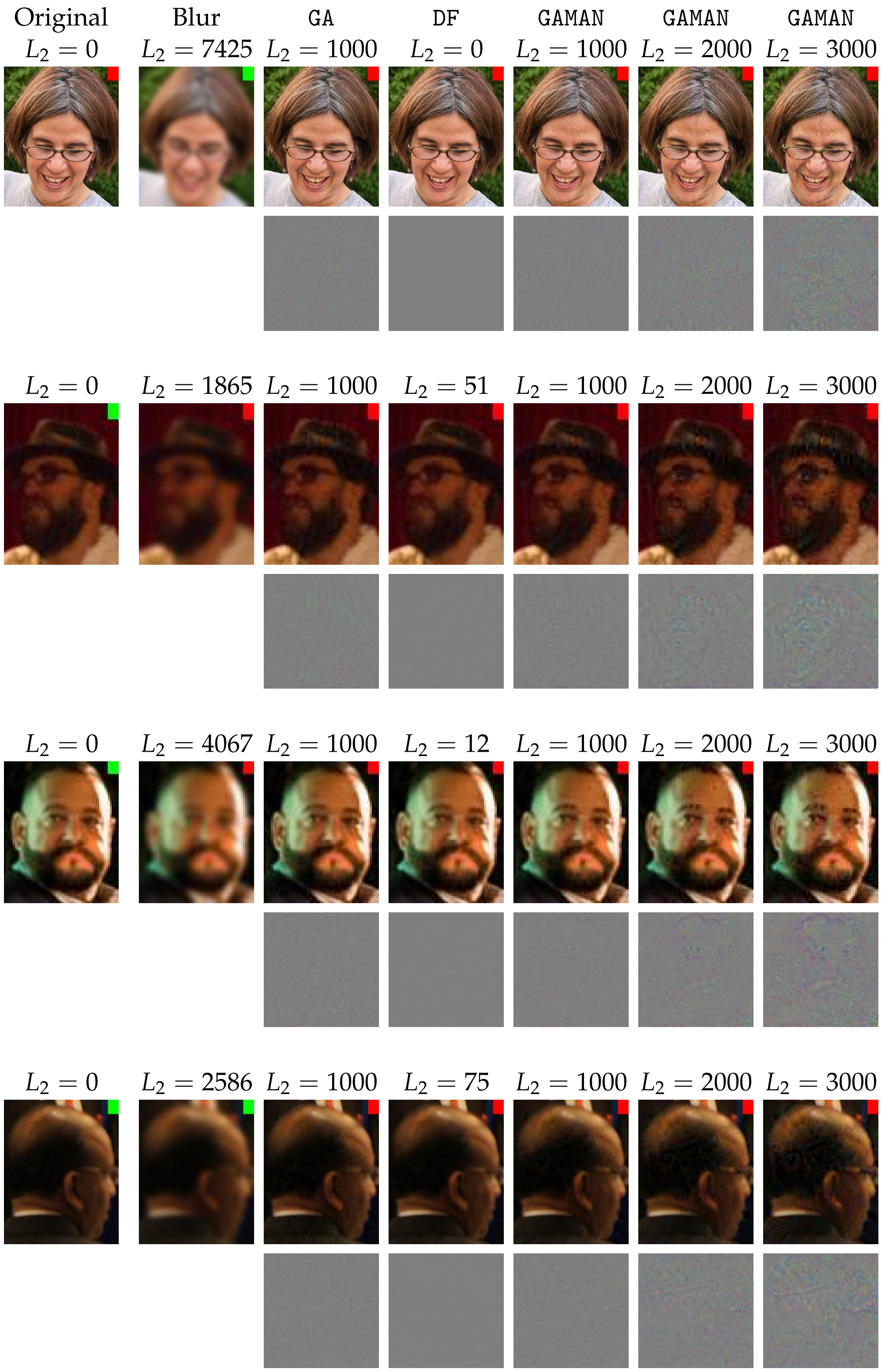


Figure 7.5: Extension of Figure 7.3. Randomly chosen perturbed images after Proc and the corresponding GoogleNet predictions (green for correct, red for wrong). Perturbations are visualised with gray background. Perturbations may be too small to be visible when printed; zoom in in electronic version for better visibility.

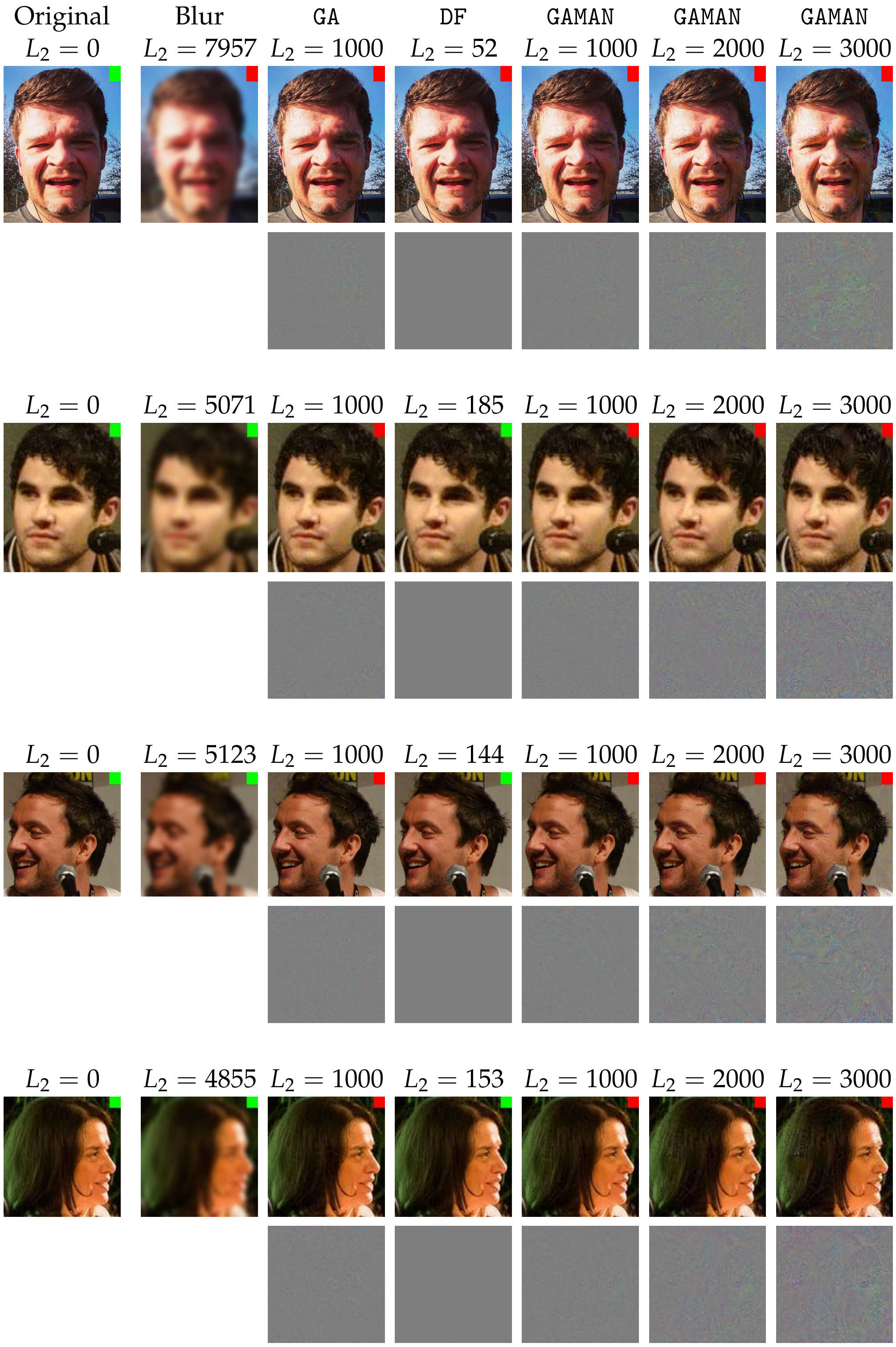


Figure 7.6: More examples. See Figure 7.5

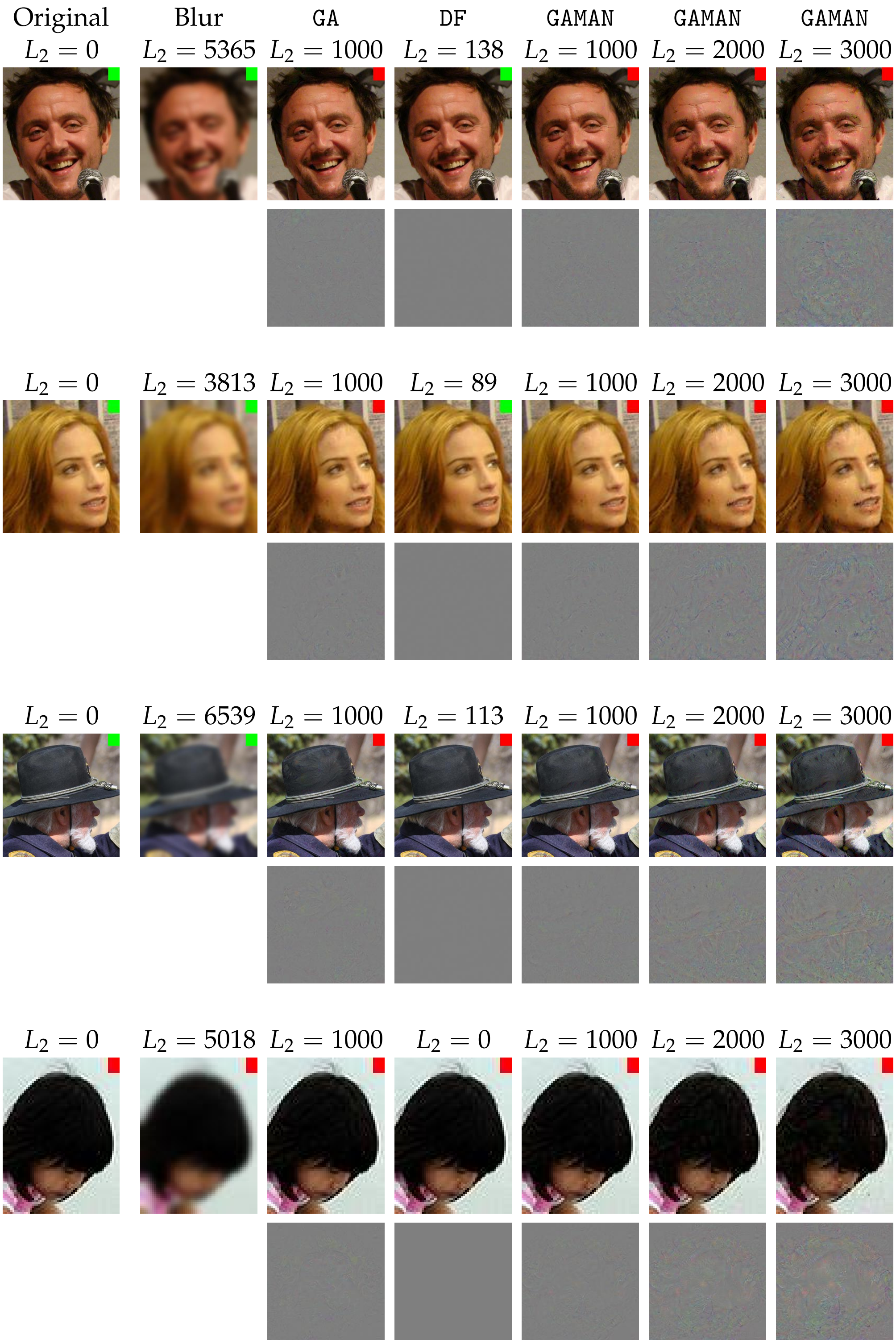


Figure 7.7: More examples. See Figure 7.5

8 TOWARDS REVERSE-ENGINEERING BLACK-BOX NEURAL NETWORKS

Many deployed learned models are black boxes: given input, returns output. Limited access limits manipulator's knowledge on the model – e.g. architecture, optimisation procedure, or training data. This chapter shows that such attributes of neural networks can be exposed from a sequence of queries. This has multiple implications. On the one hand, our work exposes the vulnerability of black-box neural networks to different types of attacks – we show that the revealed internal information helps generate more effective adversarial examples against the black box model. On the other hand, this technique can be used for better protection of private content from automatic recognition models using adversarial examples. It is actually hard to draw a line between white box and black box models.

The chapter is based on the paper Oh *et al.* (2018). As the first author, Seong Joon Oh has conducted all the experiments and has written the conference version manuscript.

8.1 INTRODUCTION

Black-box models take a sequence of query inputs, and return corresponding outputs, while keeping internal states such as model architecture hidden. They are deployed as black boxes usually on purpose – for protecting intellectual properties or privacy-sensitive training data. Our work aims at inferring information about the internals of black box models – ultimately turning them into white box models. Such a reverse-engineering of a black box model has many implications. On the one hand, it has legal implications to intellectual properties (IP) involving neural networks – internal information about the model can be proprietary and a key IP, and the training data may be privacy sensitive. Disclosing hidden details may also render the model more susceptible to attacks from adversaries. On the other hand, gaining information about a black-box model can be useful in other scenarios. E.g. there has been work on utilising adversarial examples for protecting private regions (e.g. faces) in photographs from automatic recognisers (Chapter 7). In such scenarios, gaining more knowledge on the recognisers will increase the chance of protecting one's privacy. Either way, it is a crucial research topic to investigate the type and amount of information that can be gained from a black-box access to a model. We make a first step towards understanding the connection between white box and black box approaches – which were previously thought of as distinct classes.

We introduce the term "model attributes" to refer to various types of information about a trained neural network model. We group them into three types: (1) architecture (e.g. type of non-linear activation), (2) optimisation process (e.g. SGD or ADAM?), and (3) training data (e.g. which dataset?). We approach the problem as a standard supervised learning task *applied over models*. First, collect a diverse set of white-box models ("meta-training set") that are expected to be similar to the target black box at least to a certain extent. Then, over the collected meta-training set, train another model ("metamodel") that takes a model as input and returns the corresponding model attributes as output. Importantly, since we want to predict attributes at test time for black-box models, the only information available for attribute prediction is the query input-output pairs. As we will see in the experiments, such input-output pairs allow to predict model attributes surprisingly well. In summary, we contribute:

- Investigation of the type and amount of internal information about the black-box model that can be extracted from querying;
- Novel metamodel methods that not only reason over outputs from static query inputs, but also actively optimise query inputs that can extract more information;
- Study of factors like size of the meta-training set, quantity and quality of queries, and the dissimilarity between the meta-training models and the test black box (generalisability);
- Empirical verification that revealed information leads to greater susceptibility of a black-box model to an adversarial example based attack.

8.2 METAMODELS

We want to find out the type and amount of internal information about a black-box model that can be revealed from a sequence of queries. We approach this by first building metamodels for predicting model attributes, and then evaluating their performance on black-box models. Our main approach, metamodel, is described in Figure 8.1. In a nutshell, the metamodel is a classifier of classifiers. Specifically, The metamodel submits n query inputs $\left[x^i\right]_{i=1}^n$ to a black box model f; the metamodel takes corresponding model outputs $\left[f(x^i)\right]_{i=1}^n$ as an input, and returns predicted model attributes as output. As we will describe in detail, the metamodel not only learns to infer model attributes from query outputs from a static set of inputs, but also searches for query inputs that are designed to extract greater amount of information from the target models.

In this section, our main methods are introduced in the context of MNIST digit classifiers. While MNIST classifiers are not fully representative of *generic* learned models, they have a computational edge: it takes only five minutes to train each of them with reasonable performance. We could thus prepare a diverse set of 11k MNIST classifiers within 40 GPU days for the meta-training and evaluation of our

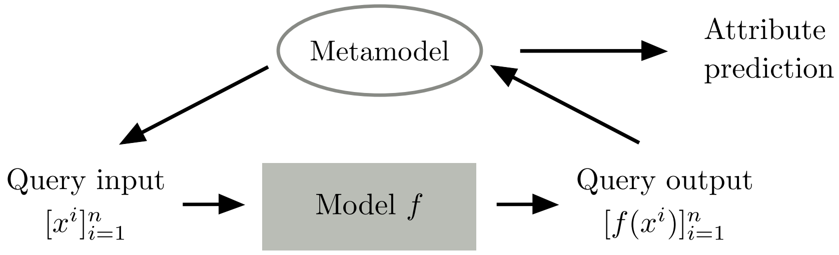


Figure 8.1: Overview of our approach.

metamodels. We stress, however, that the proposed approach is generic with respect to the task, data, and the type of models. We also focus on 12 model attributes (Table 8.1) that cover hyperparameters for common neural network MNIST classifiers, but again the range of predictable attributes are not confined to this list.

8.2.1 Collecting a dataset of classifiers

We need a dataset of classifiers to train and evaluate metamodels. We explain how `MNIST-NETS` has been constructed, a dataset of 11k MNIST digit classifiers; the procedure is task and data generic.

Base network skeleton. Every model in `MNIST-NETS` shares the same convnet skeleton architecture:

$$N \text{ conv blocks} \rightarrow M \text{ fc blocks} \rightarrow 1 \text{ linear classifier.} \tag{8.1}$$

Each conv block has the following structure:

$$\text{ks} \times \text{ks convolution} \rightarrow \text{optional } 2 \times 2 \text{ max-pooling} \rightarrow \text{non-linear activation,} \tag{8.2}$$

where ks (kernel size) and the activation type are to be chosen. Each fc block has the structure:

$$\text{linear mapping} \rightarrow \text{non-linear activation} \rightarrow \text{optional dropout.} \tag{8.3}$$

This convnet structure already covers many LeNet (LeCun *et al.*, 1998) variants, one of the best performing architectures on MNIST[20].

Increasing diversity. In order to learn generalisable features, the metamodel needs to be trained over a diverse set of models. The base architecture described above already has several free parameters like the number of layers (N and M), the existence of dropout or max-pooling layers, or the type of non-linear activation.

Apart from the architectural hyperparameters, we increase diversity along two more axes – optimisation process and the training data. Along the optimisation axis,

[20] http://yann.lecun.com/exdb/mnist/

	Code	Attribute	Values
Architecture	act	Activation	ReLU, PReLU, ELU, Tanh
	drop	Dropout	Yes, No
	pool	Max pooling	Yes, No
	ks	Conv ker. size	3, 5
	#conv	#Conv layers	2, 3, 4
	#fc	#FC layers	2, 3, 4
	#par	*#Parameters*	$2^{14}, \cdots, 2^{21}$
	ens	Ensemble	Yes, No
Opt.	alg	Algorithm	SGD, ADAM, RMSprop
	bs	Batch size	64, 128, 256
Data	split	Data split	All$_0$, Half$_{0/1}$, Quarter$_{0/1/2/3}$
	size	*Data size*	All, Half, Quarter

Table 8.1: MNIST classifier attributes. *Italicised* attributes are derived from other attributes.

we vary optimisation algorithm (SGD, ADAM, or RMSprop) and the training batch size (64, 128, 256). We also consider training MNIST classifiers on either on the entire MNIST training set (All$_0$, 60k), one of the two disjoint halves (Half$_{0/1}$, 30k), or one of the four disjoint quarters (Quarter$_{0/1/2/3}$, 15k).

See Table 8.1 for the comprehensive list of 12 model attributes altered in `MNIST-NETS`. The number of trainable parameters (#par) and the training data size (size) are not directly controlled but derived from the other attributes. We also augment `MNIST-NETS` with ensembles of classifiers (ens), whose procedure will be described later.

Sampling and training. The number of all possible combinations of controllable options in Table 8.1 is 18,144. We also select random seeds that control the initialisation and training data shuffling from $\{0, \cdots, 999\}$, resulting in 18,144,000 unique models. Training such a large number of models is intractable; we have sampled (without replacement) and trained 10,000 of them. All the models have been trained with learning rate 0.1 and momentum 0.5 for 100 epochs. It takes around 5 minutes to train each model on a GPU machine (GeForce GTX TITAN); training of 10k classifiers has taken 40 GPU days.

Pruning and augmenting. In order to make sure that `MNIST-NETS` realistically represents commonly used MNIST classifiers, we have pruned low-performance classifiers (validation accuracy$<$ 98%), resulting in 8,582 classifiers. Ensembles of trained classifiers have been constructed by grouping the identical classifiers (modulo random seed). Given t identical ones, we have augmented `MNIST-NETS` with $2, \cdots, t$ combinations. The ensemble augmentation has resulted in 11,282 final models. See Table 8.2 for statistics of attributes – due to large sample size all the attributes are evenly covered. The corresponding classification accuracies also do not correlate much with the attributes. We thus make sure that the classification accuracy alone cannot be a strong cue for predicting attributes.

	arch/act				arch/drop		arch/pool		arch/ks		arch/#conv			arch/#fc		
	Tanh	PReLU	ReLU	ELU	Yes	No	Yes	No	5	3	2	3	4	2	3	4
Ratio	24.8	24.9	25.3	25.1	49.8	50.3	49.9	50.2	50.3	49.7	34.0	33.4	32.7	33.1	33.5	33.4
max	99.4	99.4	99.5	99.4	99.5	99.4	99.4	99.5	99.5	99.4	99.4	99.4	99.5	99.4	99.4	99.5
median	98.6	98.7	98.7	98.7	98.7	98.6	98.7	98.5	98.7	98.6	98.6	98.7	98.7	98.7	98.6	98.6
mean	98.6	98.7	98.7	98.7	98.7	98.6	98.7	98.6	98.7	98.6	98.6	98.7	98.7	98.7	98.6	98.6
min	98.0	98.0	98.0	98.0	98.0	98.0	98.0	98.0	98.0	98.0	98.0	98.0	98.0	98.0	98.0	98.0

	opt/alg			opt/bs			data/size		
	RMSprop	ADAM	SGD	64	128	256	all	half	quarter
Ratio	33.8	32.5	33.7	32.9	33.6	33.7	14.8	28.5	56.8
max	99.2	99.4	99.5	99.3	99.4	99.5	99.5	99.3	99.1
median	98.6	98.7	98.7	98.6	98.7	98.7	99.0	98.8	98.5
mean	98.6	98.7	98.7	98.6	98.7	98.6	98.9	98.8	98.5
min	98.0	98.0	98.0	98.0	98.0	98.0	98.0	98.0	98.0

Table 8.2: Distribution of attributes in `MNIST-NETS`, and attribute-wise classification performance (on MNIST validation set).

Train-eval splits. Attribute prediction can get arbitrarily easy by including the black-box model (or similar ones) in the meta-training set. We introduce multiple splits of `MNIST-NETS` with varying requirements on generalization. Unless stated otherwise, every split has $5,000$ training (meta-training), $1,000$ testing (black box), and $5,282$ leftover models.

The Random (R) split randomly (uniform weights) assigns training and test splits, respectively. Under the R split, the training and test models come from the same distribution. We introduce harder Extrapolation (E) splits. We separate a few attributes between the training and test splits. They are designed to simulate more difficult domain gaps when the meta-training models are significantly different from the black box. Specific examples of E splits will be shown in §8.3.1.

8.2.2 Training metamodels

The metamodel predicts the attribute of a black-box model g in the test split by submitting n query inputs and observing the outputs. It is trained over meta-training models f in the training split ($f \sim \mathcal{F}$). We propose three approaches for the metamodels – we collectively name them `kennen`[21]. See Figure 8.2 for an overview.

`kennen-o`: **reason over output.** `kennen-o` first selects a fixed set of queries $[x^i]_{i=1\cdots n}$ from a dataset. Both during training and testing, always these queries are submitted. `kennen-o` learns a classifier m_θ to map from the order-sensitively concatenated n query outputs, $[f(x^i)]_{i=1\cdots n}$ ($n \times 10$ dim for MNIST), to the simultaneous prediction

[21] *kennen* means "to know" in German, and "to dig out" in Korean.

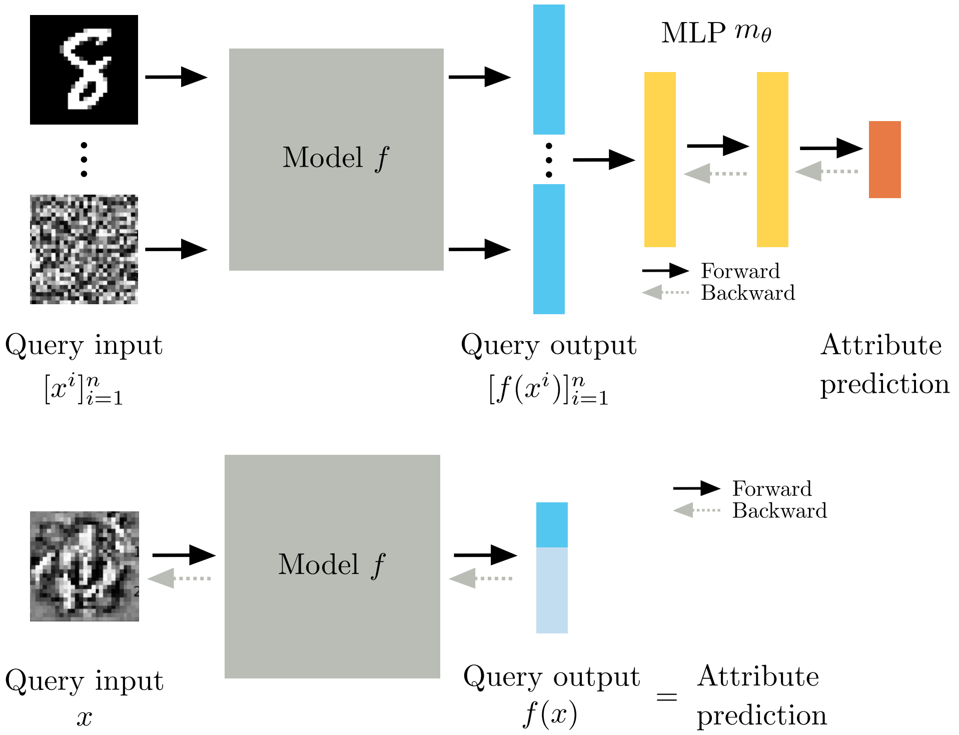


Figure 8.2: Training procedure for metamodels kennen-o (top) and kennen-i (bottom).

of 12 attributes in f. The training objective is:

$$\min_\theta \mathop{\mathbb{E}}_{f\sim\mathcal{F}} \left[\sum_{a=1}^{12} \mathcal{L}\left(m_\theta^a \left([f(x^i)]_{i=1}^n \right), y^a \right) \right] \tag{8.4}$$

where $\mathcal{F}$ is the distribution of meta-training models, y^a is the ground truth label of attribute a, and $\mathcal{L}$ is the cross-entropy loss. With the learned parameter $\tilde{\theta}$, $m_{\tilde{\theta}}^a \left([g(x^i)]_{i=1}^n \right)$ gives the prediction of attribute a for the black box g.

In our experiments, we model the classifier m_θ via multilayer perceptron (MLP) with two hidden layers with 1000 hidden units. The last layer consists of 12 parallel linear layers for a simultaneous prediction of the attributes. In our preliminary experiments, MLP has performed better than the linear classifiers. The optimisation problem in Equation 8.4 is solved via SGD by approximating the expectation over $f \sim \mathbb{F}$ by an empirical sum over the training split classifiers for 200 epochs.

For query inputs, we have used a random subset of n images from the validation set (both for MNIST and ImageNet experiments). The performance is not sensitive to the choice of queries. Next methods (kennen-i/io) describe how to actively craft query inputs, potentially outside the natural image distribution.

Note that kennen-o can be applied to any type of model (e.g. non-neural networks) with any output structure, as long as the output can be embedded in an Euclidean space. We will show that this method can effectively extract information from f even if the output is a top-k ranking.

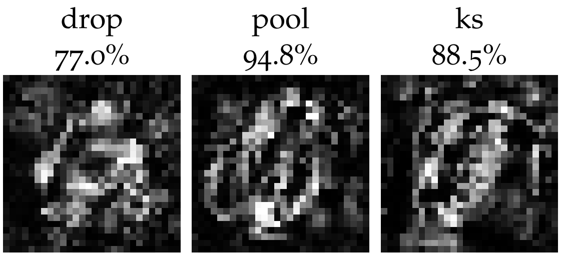


Figure 8.3: kennen-i crafted inputs and their performances. E.g. with 94.8% chance a black box will predict the middle image as "1" if it has max-pooling layers; "0" otherwise.

kennen-i**: craft input.** kennen-i crafts a *single* query input $\tilde{x}$ over the meta-training models that is trained to re-purpose a digit classifier f into a model attribute classifier for a *single* attribute a. The crafted input drives the classifier to leak internal information via digit prediction. The learned input is submitted to the test black-box model g, and the attribute is predicted by reading off its digit prediction $g(\tilde{x})$. For example, kennen-i for max-pooling layer prediction crafts an input x that is predicted as "1" for generic MNIST digit classifiers with max-pooling layers and "0" for ones without. See Figure 8.3 for visual examples.

We describe in detail how kennen-i learns this input. The training objective is:

$$\min_{x:\text{ image}} \mathop{\mathbb{E}}_{f\sim\mathcal{F}} \left[\mathcal{L}\left(f(x), y^a\right)\right] \tag{8.5}$$

where $f(x)$ is the 10-dimensional output of the digit classifier f. The condition x : image ensures the input stays a valid image $x \in [0,1]^D$ with image dimension D. The loss $\mathcal{L}$, together with the attribute label y^a of f, guides the digit prediction $f(x)$ to reveal the attribute a instead. Note that the optimisation problem is identical to the training of digit classifiers except that the ground truth is the attribute label rather than the digit label, that the loss is averaged over the models instead of the images, and that the input x instead of the model f is optimised. With the learned query input $\tilde{x}$, the attribute for the black box g is predicted by $g(\tilde{x})$. In particular, we do not use gradient information from g.

We initialise x with a random sample from the MNIST validation set (random noise or uniform gray initialisation gives similar performances), and run SGD for 200 epochs. For each iteration x is truncated back to $[0,1]^D$ to enforce the constraint.

While being simple and effective, kennen-i can only predict a single attribute at a time, and cannot predict attributes with more than 10 classes (for digit classifiers). kennen-io introduced below overcomes these limitations. kennen-i may also be unrealistic when the exploration needs to be stealthy: it submits unnatural images to the system. Also unlike kennen-o, kennen-i requires end-to-end differentiability of the training models $f \sim \mathcal{F}$, although it still requires only black-box access to test models g.

kennen-io**: combined approach.** We overcome the drawbacks of kennen-i that it can only predict one attribute at a time and that the number of predictable classes by attaching an additional interpretation module on top of the output. Our final method kennen-io combines kennen-i and kennen-o approaches: both input generator and output interpreters are used. Being able to reason over multiple query outputs via MLP layers, kennen-io supports the optimisation of multiple query inputs as well.

Specifically, the kennen-io training objective is given by:

$$\min_{[x^i]_{i=1}^n:\text{ images}} \min_{\theta} \mathop{\mathbb{E}}_{f\sim\mathcal{F}} \left[\sum_{a=1}^{12} \mathcal{L}\left(m_\theta^a \left([f(x^i)]_{i=1}^n \right), y^a \right) \right]. \tag{8.6}$$

Note that the formulation is identical to that for kennen-o (Equation 8.4), except that the second minimisation problem regarding the query inputs is added. With learned parameters $\tilde{\theta}$ and $[\tilde{x}^i]_{i=1}^n$, the attribute a for the black box g is predicted by $m_{\tilde{\theta}}^a\left([g(\tilde{x}^i)]_{i=1}^n\right)$. Again, we require end-to-end differentiability of meta-training models f, but only the black-box access for the test model g.

To improve stability against covariate shift, we initialise m_θ with kennen-o for 200 epochs. Afterwards, gradient updates of $[x^i]_{i=1}^n$ and θ alternate every 50 epochs, for 200 additional epochs.

8.3 REVERSE-ENGINEERING BLACK-BOX MNIST DIGIT CLASSIFIERS

We have introduced a procedure for constructing a dataset of classifiers (MNIST-NETS) as well as novel metamodels (kennen variants) that learn to extract information from black-box classifiers. In this section, we evaluate the ability of kennen to extract information from black-box MNIST digit classifiers. We measure the *class-balanced* attribute prediction accuracy for each attribute a in the list of 12 attributes in Table 8.1.

Attribute prediction. See Table 8.3 for the main results of our metamodels, kennen-o/i/io, on the Random split. Unless stated otherwise, metamodels are trained with 5,000 training split classifiers.

Given $n = 100$ queries with probability output, kennen-o already performs far above the random chance in predicting 12 diverse attributes (73.4% versus 34.9% on average); neural network output indeed contains rich information about the black box. In particular, the presence of dropout (94.6%) or max-pooling (94.9%) has been predicted with high precision. As we will see in §8.3.2, outputs of networks trained with dropout layers form clusters, explaining the good prediction performance.

It is surprising that optimisation details like algorithm (71.8%) and batch size (50.4%) can also be predicted well above the random chance (33.3% for both). We observe that the training data attributes are also predicted with high accuracy (71.8% and 90.0% for size and split).

Comparing methods kennen-o/i/io. Table 8.3 shows the comparison of kennen-o/i/io. kennen-i has a relatively low performance (average 52.7%), but kennen-i relies on a cheap resource: 1 query with single-label output. kennen-i is also performant at

		architecture								optim		data		
Method	Output	act	drop	pool	ks	#conv	#fc	#par	ens	alg	bs	size	split	avg
Chance	-	25.0	50.0	50.0	50.0	33.3	33.3	12.5	50.0	33.3	33.3	33.3	14.3	34.9
`kennen-o`	prob	80.6	94.6	94.9	84.6	67.1	77.3	41.7	54.0	71.8	50.4	73.8	90.0	73.4
`kennen-o`	ranking	63.7	93.8	90.8	80.0	63.0	73.7	44.1	**62.4**	65.3	47.0	66.2	86.6	69.7
`kennen-o`	bottom-1	48.6	80.0	73.6	64.0	48.9	63.1	28.7	52.8	53.6	41.9	45.9	51.4	54.4
`kennen-o`	top-1	31.2	56.9	58.8	49.9	38.9	33.7	19.6	50.0	36.1	35.3	33.3	30.7	39.5
`kennen-i`	top-1	43.5	77.0	94.8	88.5	54.5	41.0	32.3	46.5	45.7	37.0	42.6	29.3	52.7
`kennen-io`	score	**88.4**	**95.8**	**99.5**	**97.7**	**80.3**	**80.2**	**45.2**	60.2	**79.3**	**54.3**	**84.8**	**95.6**	**80.1**

Table 8.3: Comparison of metamodel methods. See Table 8.1 for the full names of attributes. 100 queries are used for every method below, except for `kennen-i` which uses a single query. The "Output" column shows the output representation: "prob" (vector of probabilities for each digit class), "ranking" (a sorted list of digits according to their likelihood), "top-1" (most likely digit), or "bottom-1" (least likely digit).

predicting the kernel size (88.5%) and pooling (94.8%), attributes that are closely linked to spatial structure of the input. We conjecture `kennen-i` is relatively effective for such attributes. `kennen-io` is superior to `kennen-o/i` for all the attributes with average accuracy 80.1%.

Factor analysis. We examine potential factors that contribute to the successful prediction of black box internal attributes. We measure the prediction accuracy of our metamodels as we vary (1) the number of meta-training models, (2) the number of queries, and (3) the quality of query output.

Number of training models. We have trained `kennen-o` with different number of the meta-training classifiers, ranging from 100 to 5,000. See Figure 8.4 (left) for the trend. We observe a diminishing return, but also that the performance has not saturated – collecting larger meta-training set will improve the performance.

number of queries. See Figure 8.4 (middle) for the `kennen-o` performance against the number of queries with probability output. The average performance saturates after $\sim$ 500 queries. On the other hand, with only $\sim$ 100 queries, we already retrieve ample information about the neural network.

Quality of output. Many black-box models return top-k ranking (e.g. Facebook face recogniser), or single-label output. We represent top-k ranking outputs by assigning exponentially decaying probabilities up to k digits and a small probability ϵ to the remaining.

See Table 8.3 for the `kennen-o` performance comparison among 100 probability, top-10 ranking, bottom-1, and top-1 outputs, with average accuracies 73.4%, 69.7%, 54.4%, and 39.5%, respectively. While performance drops with coarser outputs, when compared to random chance (34.9%), 100 single-label bottom-1 outputs already leak

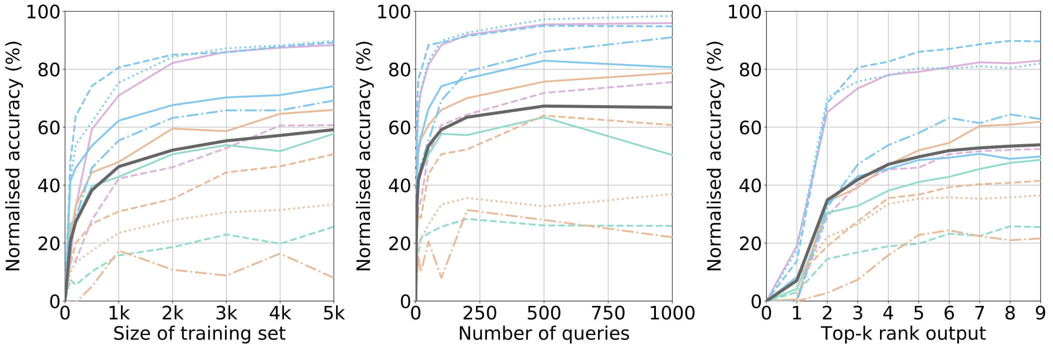


Figure 8.4: kennen-o performance against the size of meta-training set (left), number of queries (middle), and quality of queries (right). Unless stated otherwise, we use 100 probability outputs and 5k models to train kennen-o. Each curve is linearly scaled such that random chance (0 training data, 0 query, or top-0) performs 0%, and the perfect predictor performs 100%. Legends for curves are given in Figure 8.5.

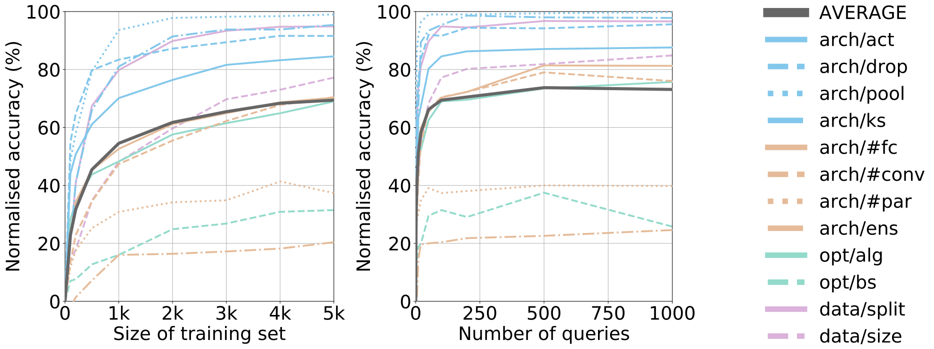


Figure 8.5: Performance of kennen-io with different number of queries (Left) and size of training set (Right). The curves are linearly scaled per attribute such that random chance performs 0%, and perfect predictor performs 100%.

a great amount of information about the black box (54.4%). It is also notable that bottom-1 outputs contain much more information than do the top-1 outputs; note that for high-performance classifiers top-1 predictions are rather uniform across models and thus have much less freedom to leak auxiliary information. Figure 8.4 (right) shows the interpolation from top-1 to top-10 (i.e. top-9) ranking. We observe from the jump at $k = 2$ that the second likely predictions (top-2) contain far more information than the most likely ones (top-1). For $k \geq 3$, each additional output label exhibits a diminishing return.

More kennen-io results. We present additional kennen-io results. See Figure 8.5. Similarly for kennen-o, kennen-io shows a diminishing return as the number of training models and the number of queries increase. While the performance saturates with $1,000$ queries, it does not fully saturate with $5,000$ training samples.

			kennen-	
Split	Train	Test	o	io
R	-	-	100	100
E-#conv	2,3	4	87.5	92.0
E-#conv-#fc	2,3	4	77.1	80.7
E-alg	SGD,ADAM	RMSprop	83.0	88.5
E-alg-bs	64,128	256	64.2	70.0
E-split	Quarter$_{0/1}$	Quarter$_{2/3}$	83.5	89.3
E-size	Quarter	Half,All	81.7	86.8
Chance	-	-	0.0	0.0

Table 8.4: Normalised accuracies (see text) of kennen-o and kennen-io on R and E splits. We denote E-split with splitting attributes *attr1* and *attr2* as "E-*attr1*-*attr2*". Splitting criteria are also shown. When there are two splitting attributes, the first attribute inherits the previous row criteria.

8.3.1 What if the black-box is quite different from meta-training models?

So far we have seen results on the Random (R) split. In realistic scenarios, the meta-training model distribution may not be fully covering possible black box models. We show how damaging such a scenario is through Extrapolation (E) split experiments.

Evaluation. E-splits split the training and testing models based on one or more attributes (§8.2.1). For example, we may assign shallower models (#layers ≤ 10) to the training split and deeper ones (#layers >10) to the testing split. In this example, we refer to #layers as the *splitting attribute*. Since for an E-split, some classes of the splitting attributes have zero training examples, we only evaluate the prediction accuracies over the non-splitting attributes. When the set of splitting attributes is $\tilde{A}$, a subset of the entire attribute set A, we define *E-split accuracy* or E.Acc$(\tilde{A})$ to be the mean prediction accuracy over the non-splitting attributes $A \setminus \tilde{A}$. For easier comparison, we report the *normalised accuracy* (N.Acc) that shows the how much percentage of the R-split accuracy is achieved in the E-split setup on the non-splitting attributes $A \setminus \tilde{A}$. Specifically:

$$\text{N.Acc}(\tilde{A}) = \frac{\text{E.Acc}(\tilde{A}) - \text{Chance}(\tilde{A})}{\text{R.Acc}(\tilde{A}) - \text{Chance}(\tilde{A})} \times 100\% \tag{8.7}$$

where R.Acc$(\tilde{A})$ and Chance$(\tilde{A})$ are the means of the R-split and Chance-level accuracies over $A \setminus \tilde{A}$. Note that N.Acc is 100% if the E-split performance is at the level of R-split and 0% if it is at chance level.

Results. The normalised accuracies for R-split and multiple E-splits are presented in Table 8.4. We consider three axes of choices of splitting attributes for the E-split:

architecture (#conv and #fc), optimisation (alg and bs), and data (size). For example, "E-#conv-#fc" row presents results when metamodel is trained on shallower nets (2 or 3 conv/fc layers each) compared to the test black box model (4 conv and fc layers each).

Not surprisingly, E-split performances are lower than R-split ones (N.Acc $<$ 100%); it is advisable to cover all the expected black-box attributes during meta-training. Nonetheless, E-split performances of `kennen-io` are still far above the chance level (N.Acc $\geq 70\% \gg 0\%$); failing to cover a few attributes during meta-training is not too damaging.

Comparing `kennen-o` and `kennen-io` for their generalisability, we observe that `kennen-io` consistently outperforms `kennen-o` under severe extrapolation (around 5 pp better N.Acc). It is left as a future work to investigate the intriguing fact that utilising out-of-domain query inputs improves the generalisation of metamodel.

8.3.2 Why and how does metamodel work?

It is surprising that metamodels can extract inner details with great precision and generalisability. This section provides a glimpse of *why* and *how* this is possible via metamodel input and output analyses. Full answers to those questions is beyond the scope of this work.

Metamodel input (t-SNE). We analyse the inputs to our metamodels (i.e. query outputs from black-box models) to convince ourselves that the inputs do contain discriminative features for model attributes. As the input is high dimensional (1000 when the number of queries is $n = 100$), we use the t-SNE (van der Maaten and Hinton, 2008) visualisation method. First, 1000 test-split (Random split) black-box models are collected. For each model, 100 query images are passed (sampled at random from MNIST validation set), resulting in 100×10 dimensional input data points. t-SNE embeds high dimensional data points onto the 2-dimensional plane such that the pairwise distances are best respected. We then colour-code the embedded data points according to the model attributes. Clusters of same-coloured points indicate highly discriminative features.

The visualisation of input data points are shown in Figures 8.6 and 8.7 for `kennen-o` and `kennen-io`, respectively. In the case of `kennen-o`, we observe that some attributes form clear clusters in the input space – e.g. Tanh in act, binary dropout attribute, and RMSprop in alg. For the other attributes, however, it seems that the clusters are too complicated to be represented in a 2-dimensional space. For `kennen-io` (Figure 8.7), we observe improved clusters for pool and ks. By submitting crafted query inputs, `kennen-io` induces query outputs to be better clustered, increasing the chance of successful prediction. Since t-SNE is sensitive to initialisation, we have run the embedding ten times with different random initialisations; the qualitative observations are largely identical.

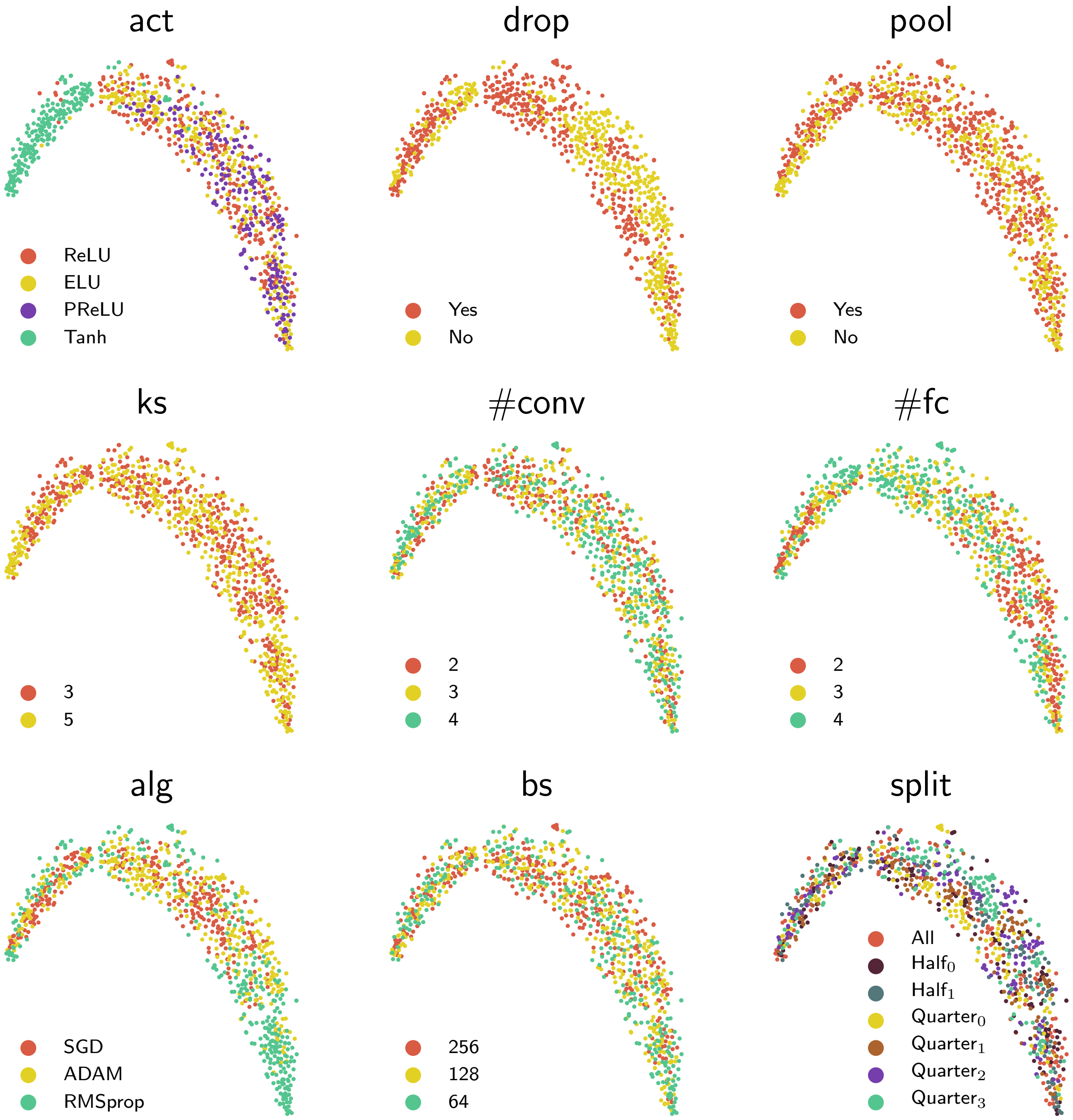


Figure 8.6: Probability query output embedded into 2-D plane via t-SNE. The same embedding is shown with different colour-coding for each attribute. These are the inputs to the `kennen-o` metamodel.

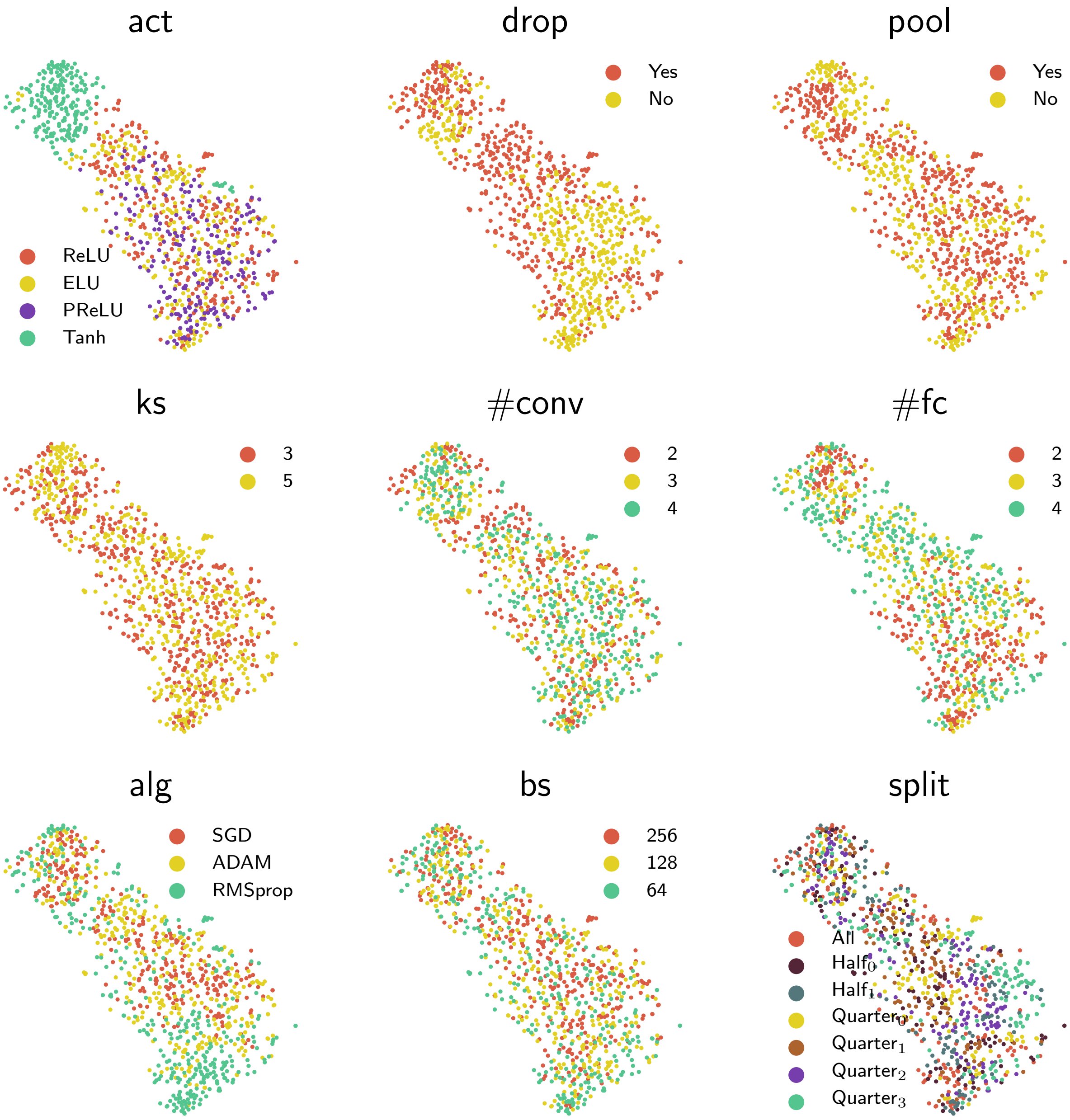


Figure 8.7: Probability query output embedded into 2-D plane via t-SNE. The same embedding is shown with different colour-coding for each attribute. These are the inputs to the `kennen-io` metamodel.

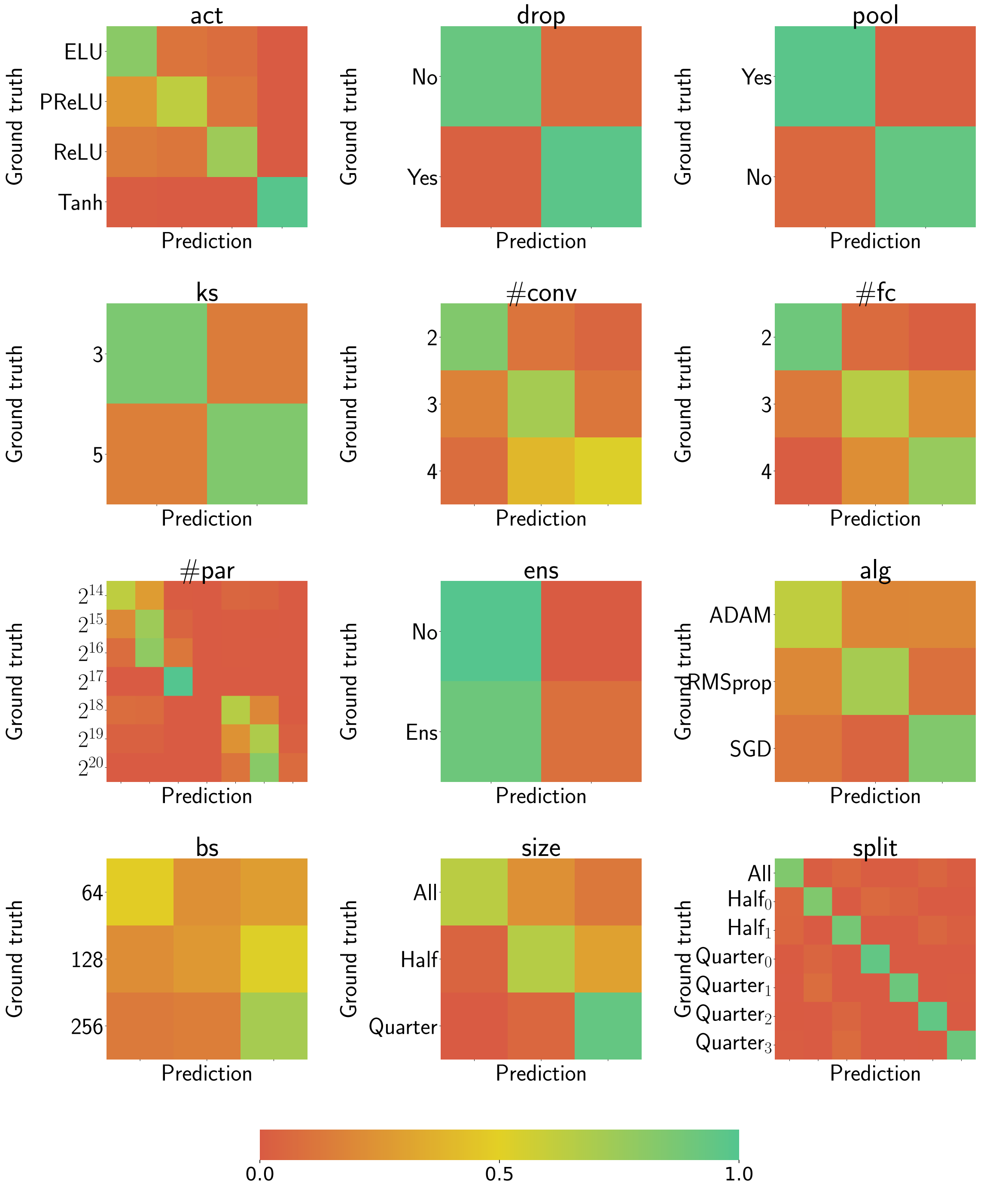


Figure 8.8: Confusion matrices for kennen-o.

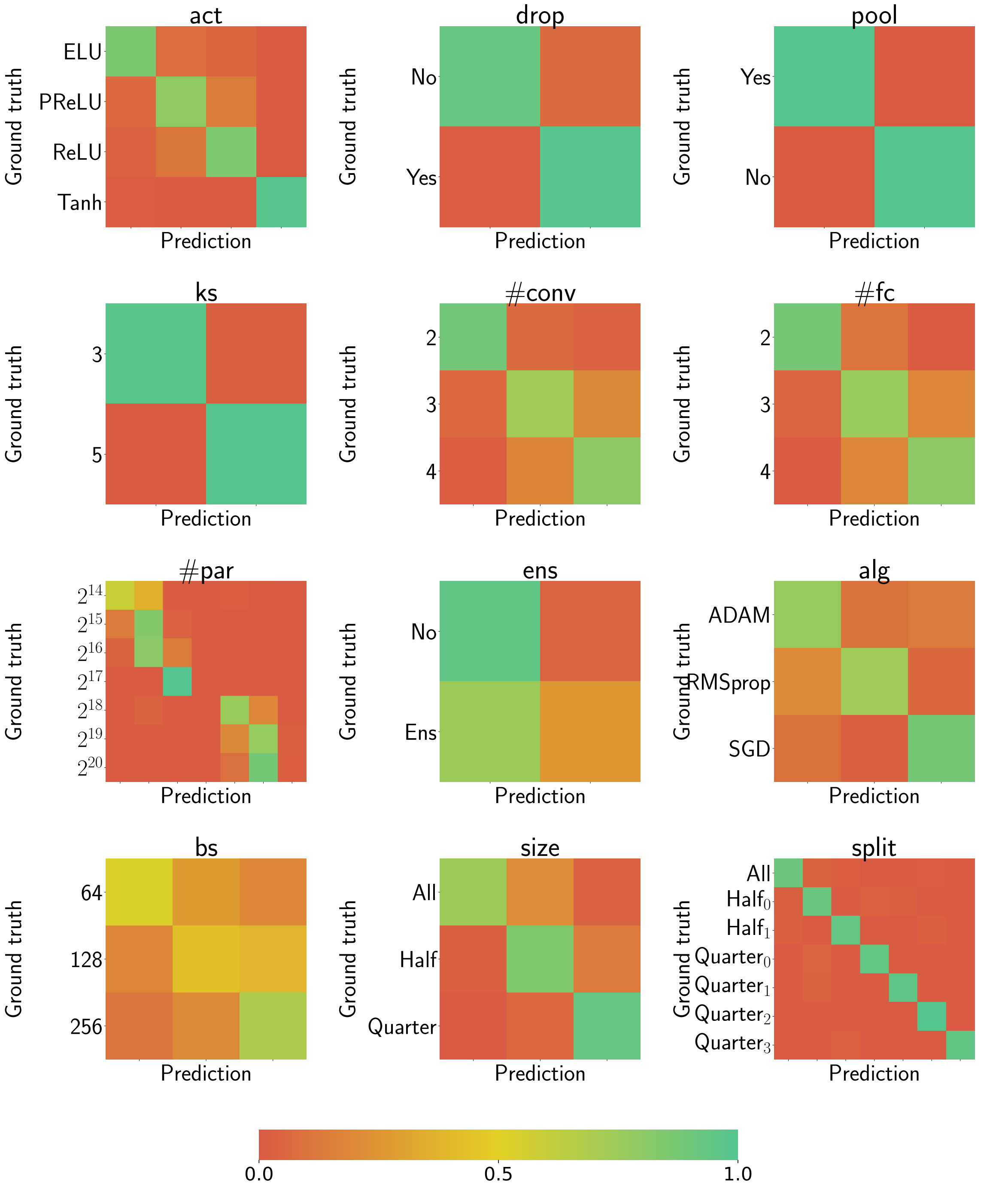


Figure 8.9: Confusion matrices for kennen-io.

Metamodel output (confusion matrix). We show confusion matrices of kennen-o/io to analyse the failure modes. See Figures 8.8 and 8.9. For `kennen-o` and `kennen-io` alike, we observe that the confusion occurs more frequently with similar classes. For attributes #conv and #fc, more confusion occurs between $(2, 3)$ or $(3, 4)$ than between $(2, 4)$. A similar trend is observed for #par and bs. This is a strong indication that (1) there exists semantic attribute information in the neural network outputs (e.g. number of layers, parameters, or size of training batch) and (2) the metamodels learn semantic information that can generalise, as opposed to merely relying on artifacts. This observation agrees with a conclusion of the extrapolation experiments in §8.3.1: the metamodels generalise.

Compared to those of `kennen-o`, `kennen-io` confusion matrices exhibit greater concentration of masses both on the correct class (diagonals) and among similar attribute classes (1-off diagonals for #conv, #fc, #par, bs, and size). The former re-confirms the greater accuracy, while the latter indicates the improved ability to extract more semantic and generalisable features from the query outputs. This, again, agrees with §8.3.1: `kennen-io` generalises better than `kennen-o`.

8.3.3 Discussion

We have verified through our novel `kennen` metamodels that black-box access to a neural network exposes much internal information. We have shown that only 100 single-label outputs already reveal a great deal about the black boxes. When the black-box classifier is quite different from the meta-training classifiers, the performance of our best metamodel – `kennen-io`– decreases; however, the prediction accuracy for black box internal information is still surprisingly high.

8.4 REVERSE-ENGINEERING AND ATTACKING IMAGENET CLASSIFIERS

While MNIST experiments are computationally cheap and a massive number of controlled experiments is possible, we provide additional ImageNet experiments for practical implications on realistic image classifiers. In this section, we use `kennen-o` introduced in §8.2 to predict a single attribute of black-box ImageNet classifiers – the architecture family (e.g. ResNet or VGG?). In this section, we go a step further to use the extracted information to attack black boxes with adversarial examples.

8.4.1 Dataset of ImageNet classifiers

It is computationally prohibitive to train $O(10k)$ ImageNet classifiers from scratch as in the previous section. We have resorted to 19 PyTorch[22] pretrained ImageNet classifiers. The 19 classifiers come from five families: **S**queezenet, **V**GG, VGG-**B**atchNorm, **R**esNet, and **D**enseNet, each with 2, 4, 4, 5, and 4 variants, respectively (Iandola

[22] https://github.com/pytorch

	S (2016)		V (2014)				B (2015)				R (2015)					D (2016)			
Description	Lightweight convnet		Conv layers followed by fc layers				VGG with batch normalisation				Very deep convnet with residual connections					ResNet with dense residual connections			
Members	v1.0	v1.1	11	13	16	19	11	13	16	19	18	34	50	101	152	121	161	169	201
Top-5 error	19.6	19.4	11.4	10.8	9.6	9.1	10.2	9.6	8.5	8.2	10.9	8.6	7.1	6.4	5.9	7.8	6.2	7.0	6.4
$\log_{10}$ #params	6.1	6.1	8.1	8.1	8.1	8.2	8.1	8.1	8.1	8.2	7.1	7.3	7.4	7.6	7.8	6.9	7.3	7.5	7.2

Table 8.5: Details of ImageNet classifiers. We describe each family **S**queezenet, **V**GG, VGG-**B**atchNorm, **R**esNet, and **D**enseNet verbally, and show key model statistics for each member in the family.

et al., 2016; Simonyan and Zisserman, 2015; Ioffe and Szegedy, 2015; He *et al.*, 2016; Huang *et al.*, 2017a). See Table 8.5 for the the summary of the 19 classifiers. We observe intra-family diversity (e.g. R) and inter-family similarity (e.g. between V and B) in terms of the top-5 validation error and the number of trainable parameters. The family prediction task is not as trivial as e.g. simply inferring the performance.

8.4.2 Classifier family prediction

We predict the classifier family (S, V, B, R, D) from the black-box query output, using the method `kennen-o`, with the same MLP architecture (§8.2). `kennen-i` and `kennen-io` have not been used for computational reasons, but can also be used in principle. We conduct 10 cross validations (random sampling of single test network from each family) for evaluation. We also perform 10 random sampling of the queries from ImageNet validation set. In total 100 random tries are averaged.

Results: compared to the random chance (20.0%), 100 queries result in high `kennen-o` performance (90.4%). With $1,000$ queries, the prediction performance is even 94.8%.

8.4.3 Attacking ImageNet classifiers

In this section we attack ImageNet classifiers with adversarial image perturbations (AIPs). We show that the knowledge about the black box architecture family makes the attack more effective.

Adversarial image perturbation (AIP). AIPs are carefully crafted additive perturbations on the input image for the purpose of misleading the target model to predict wrong labels (Goodfellow *et al.*, 2015). Among variants of AIPs, we use efficient and robust GAMAN (Chapter 7). See Figure 8.10 for examples of AIPs; the perturbation is nearly invisible.

Transferability of AIPs. Typical AIP algorithms require gradients from the target network, which is not available for a black box. Mainly three approaches for generating AIPs against black boxes have been proposed: (1) numerical gradient, (2)

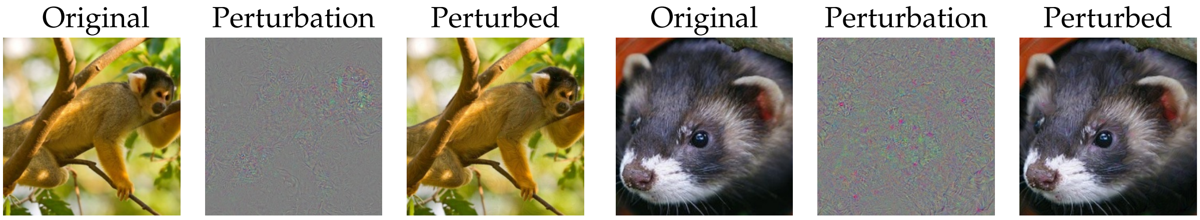


Figure 8.10: AIP for an ImageNet classifier. The perturbations are generated at $L_2 = 1 \times 10^{-4}$.

Gen	Target family				
	S	V	B	R	D
Clean	38	32	28	30	29
S	64	49	45	39	35
V	62	96	96	57	52
B	50	85	95	47	44
R	64	72	78	87	77
D	58	63	70	76	90
Ens	70	93	93	75	80

Table 8.6: Transferability of AIPs within and across families. We report the *misclassification rates*.

avatar network, or (3) transferability. We show that our metamodel strengthens the transferability based attack.

We hypothesize and empirically show that AIPs transfer better within the architecture family than across. Using this property, we first predict the family of the black box (e.g. ResNet), and then generate AIPs against a few instances in the family (e.g. ResNet101, ResNet152). The generation of AIPs against multiple targets has been proposed by Liu *et al.* (2017a), but we are the first to systemically show that AIPs generalise better within a family when they are generated against multiple instances from the same family.

We first verify our hypothesis that AIPs transfer better within a family. Within-family: we do a leave-one-out cross validation – generate AIPs using all but one instances of the family and test on the holdout. Not using the exact test black box, this gives a lower bound on the within-family performance. Across-family: still leave out one random instance from the generating family to match the generating set size with the within-family cases. We also include the use-all case (Ens): generate AIPs with one network from *each* family.

See Table 8.6 for the results. We report the *misclassification rate*, defined as 100−top-1 accuracy, on 100 random ImageNet validation images. We observe that the within-family performances dominate the across-family ones (diagonal entries versus the others in each row); if the target black box family is identified, one can generate more effective AIPs. Finally, trying to target all network ("Ens") is not as effective as focusing resources (diagonal entries).

Scenario	Generating nets	MC(%)
White box	Single white box	100.0
Family black box	GT family	86.2
Black box whitened	**Predicted family**	**85.7**
Black box	Multiple families	82.2

Table 8.7: Black-box ImageNet classifier misclassification rates (MC) for different approaches.

Metamodel enables more effective attacks. We empirically show that the reverse-engineering enables more effective attacks. We consider multiple scenarios. "White box" means the target model is fully known, and the AIP is generated specifically for this model. "Black box" means the exact target is unknown, but we make a distinction when the family is known ("Family black box").

See Table 8.7 for the misclassification rates (MC) in different scenarios. When the target is fully specified (white box), MC is 100%. When neither the exact target nor the family is known, AIPs are generated against multiple families (82.2%). When the reverse-engineering takes place, and AIPs are generated over the predicted family, attacks become more effective (85.7%). We almost reach the family-oracle case (86.2%).

8.4.4 Discussion

Our metamodel can predict architecture families for ImageNet classifiers with high accuracy. We additionally show that this reverse-engineering enables more focused attack on black-boxes.

8.5 CONCLUSION

We have presented first results on the inference of diverse neural network attributes from a sequence of input-output queries. Our novel metamodel methods, `kennen`, can successfully predict attributes related not only to the architecture but also to training hyperparameters (optimisation algorithm and dataset) even in difficult scenarios (e.g. single-label output, or a distribution gap between the meta-training models and the target black box).

Using `kennen`, we can increase the manipulator's knowledge only through a sequence of queries. This shows that manipulator's ability may have been underestimated against black boxes. From security point of view, this chapter exposes new vulnerabilities of black-box neural networks to stealing attacks and adversarial perturbation attacks. From privacy point of view, this will enable more successful identity obfuscation against black-box person recognition models.

9 EXPLOITING SALIENCY FOR OBJECT SEGMENTATION FROM IMAGE LEVEL LABELS

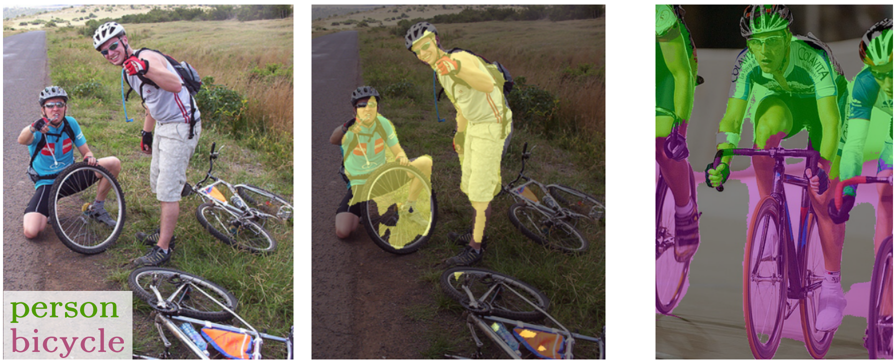
This chapter is an interlude chapter, but is connected to the general question of how to extract knowledge from a model. Using techniques to extract high-confidence object location information from an image classifier, we train a semantic segmentation network that bypasses the need for collecting expensive pixel-level annotations. Since obtaining the full extent of the objects is not possible with only a classifier, we propose using saliency as prior knowledge on the object extent. We show how to combine both information sources in order to recover 80% of the fully supervised performance, the new state of the art in weakly supervised training for pixel-wise semantic labelling.

The chapter is based on the paper Oh *et al.* (2017b). As the first author, Seong Joon Oh has conducted most of the experiments and has written the conference version of the manuscript. The weakly supervised saliency network in §9.4 and related experiments are contributed by Dr Anna Khoreva (co-author).

9.1 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 trade-offs 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).



(a) Image labels (b) Saliency (c) Our result

Figure 9.1: 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.

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 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 ~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.

9.1.1 Related work on weakly supervised semantic segmentation

The last years have seen a renewed interest on weakly supervised training. For semantic labelling, different forms of supervision have been explored: image labels (Pathak *et al.*, 2015b,a; Papandreou *et al.*, 2015; Pinheiro and Collobert, 2015; Wei *et al.*, 2015; Kolesnikov and Lampert, 2016b), points (Bearman *et al.*, 2016), scribbles (Xu *et al.*, 2015; Lin *et al.*, 2016), and bounding boxes (Dai *et al.*, 2015; Papandreou *et al.*, 2015; Khoreva *et al.*, 2017). In this work we focus on image labels as the main form of supervision.

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 by-product of a forward-pass. This can be achieved by adding a global a 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) (§9.3).

Pixel labelling from image level supervision. Initial work approached this problem by adapting multiple-instance learning (Pathak *et al.*, 2015b) and expectationmaximization techniques (Papandreou *et al.*, 2015), 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 uses CRFs to propagate the seeds across the image during training, while Qi *et al.* (2016) exploits segment proposals for this.

Most methods compared propose each a new procedure to train a semantic labelling convnet. One exception is Shimoda and Yanai (2016) which fuses at test time guided back-propagation (Springenberg *et al.*, 2015) at multiple convnet layers to generate class-wise heatmaps. They do this over a convnet trained for classification. Being based on classifier, their output masks only partially capture the object extents, as reflected in the comparatively low performance (Table 9.3).

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. The closest work to our approach is Wei *et al.* (2015), which also uses saliency as a cue to improve weakly supervised semantic segmentation. There are however a number of differences. First, they use a curriculum learning to expose the segmentation convnet first with simple images, and later with more complex ones. We do not need such curriculum, yet reach better results. Second, they use a manually crafted class-agnostic saliency method, while we use a deep learning based one (which provides better cues). Third, their training procedure uses ~ 40k additional images of the classes of interest crawled from the web; we do not use such class-specific external data. Fourth, we report significantly better results, showing in better light the potential of saliency as additional information to guide weakly supervised semantic object labelling.

The seminal work Vezhnevets *et al.* (2011) proposed to use "objectness" map from bounding boxes to guide the semantic segmentation. By using bounding boxes, these maps end up being diffuse; in contrast, our saliency map has sharp object boundaries, thus giving more precise guidance to the semantic labeller.

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 architecture 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.

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 object a person might look first (Li *et al.*, 2014b), 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 last definition in this paper. Note that this notion is class-agnostic, and refers more to the composition of the image, than the specific object category.

Like most computer vision areas, hand-crafted methods (Jiang *et al.*, 2013; Margolin *et al.*, 2013; Cheng *et al.*, 2015) have now been surpassed by convnet based approaches (Zhao *et al.*, 2015; Li *et al.*, 2016b; Li and Yu, 2016) for object saliency. In this paper we use saliency as an ingredient: improved saliency models would lead to improved results for our method. We describe in §9.5.1 our saliency model design, trained itself in a weakly supervised fashion from bounding boxes.

Semantic labelling. Even when pixel-level annotations are provided (fully supervised case), the task of semantic labelling is far from being solved. Multiple convnet architectures have been proposed, including recurrent networks (Pinheiro and Collobert, 2014), encoder-decoders (Noh *et al.*, 2015; Badrinarayanan *et al.*, 2017), up-sampling layers (Long *et al.*, 2015a), using skip layers (Bansal *et al.*, 2016), or dilated convolutions (Chen *et al.*, 2016; Yu and Koltun, 2016), to name a few. Most of them build upon classification architectures such as VGG (Simonyan and Zisserman, 2015) or ResNet (He *et al.*, 2016). For comparison with previous work, our experiments are based on the popular DeepLab (Chen *et al.*, 2016) architecture.

9.2 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 9.2), we name this architecture "Guided Segmentation". Given an image and image-level labels, the "guide labeller" module combines cues from a seeder (§9.3) and saliency (§9.4) 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 §9.3 and §9.4 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.

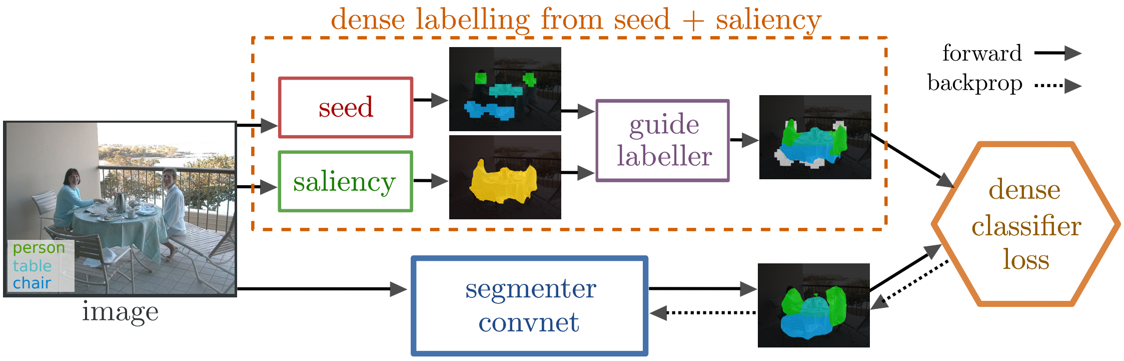


Figure 9.2: High level Guided Segmentation architecture.

9.3 FINDING GOOD SEEDS

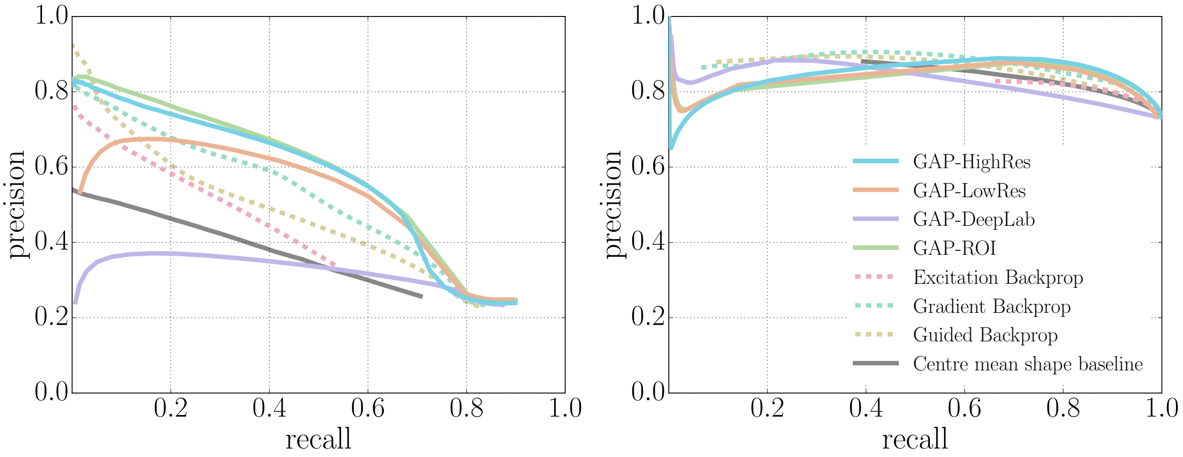
There has been recent burst of approaches to localise objects from a classifier. Some approaches rely on image gradients from a trained classifier (Simonyan *et al.*, 2014; Springenberg *et al.*, 2015; Zhang *et al.*, 2016), while others propose to train a global average pooling (GAP) based architectures 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 their empirical performances. We report experimental results of different GAP architectures (Zhou *et al.*, 2016; Kolesnikov and Lampert, 2016b; Chen *et al.*, 2016), where we show that good architectural components for a classifier or segmenter may not lead to a good GAP architecture.

9.3.1 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 9.1. `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



(a) Foreground categories

(b) Background category

Figure 9.3: Comparing seeds techniques. Precision-recall curves.

	GAP			
	-LowRes	-HighRes	-ROI	-DeepLab
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 9.1: 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.

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.

9.3.2 Empirical study

Evaluation. We evaluate each method on the *val* set of the Pascal VOC 2012 (Everingham *et al.*, 2012) segmentation benchmark. We measure the foreground and background precision-recall curves for each variant. In the foreground case, we compute the mean precision and recall over the 20 Pascal categories.

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

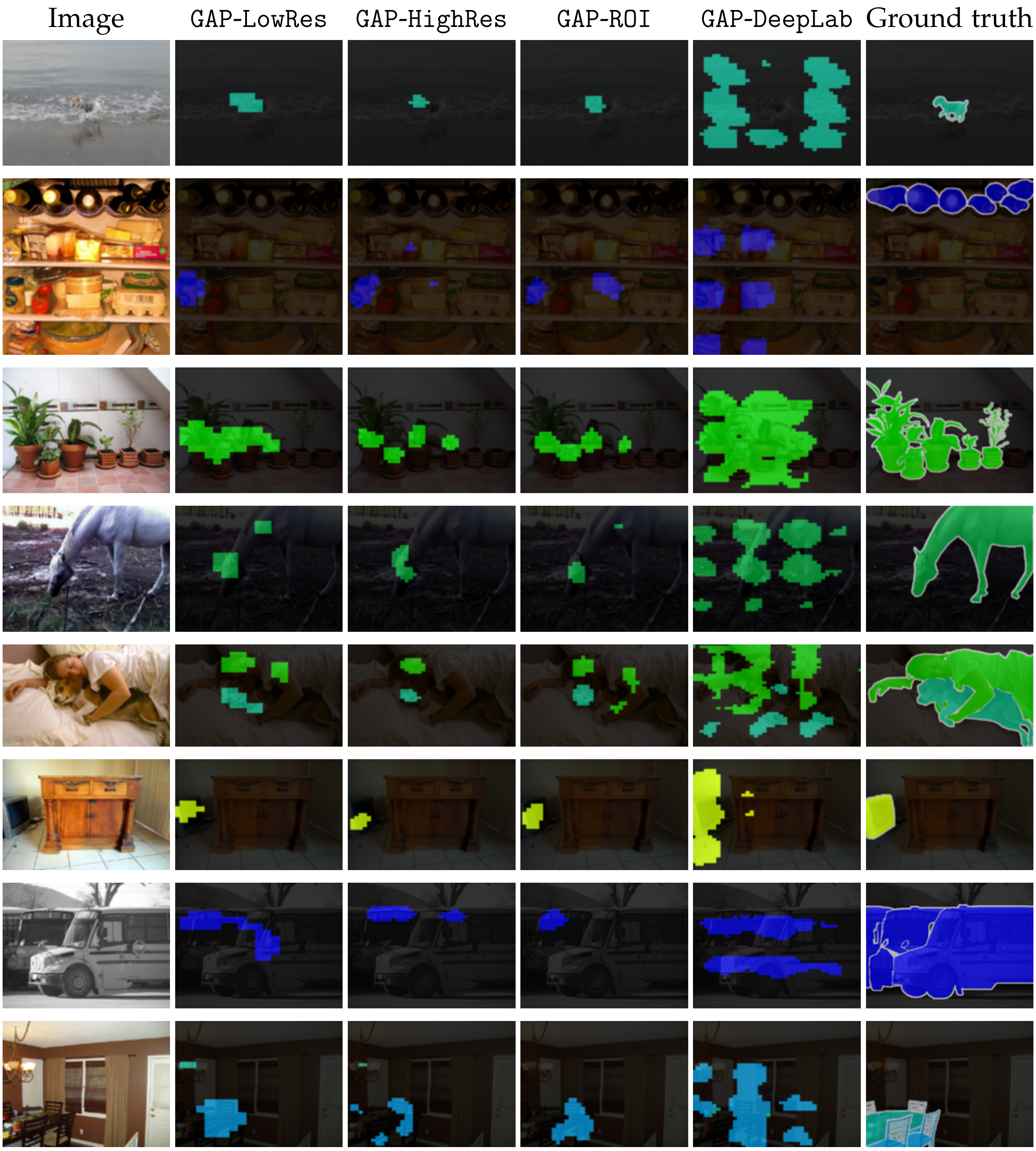


Figure 9.4: Qualitative examples of GAP output for GAP-LowRes, GAP-HighRes, GAP-DeepLab, and GAP-ROI.

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 (Hariharan *et al.*, 2011). 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 9.3 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 9.1 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.

Figure 9.4 shows example outputs of GAP variants chosen at random. All of them, except for GAP-DeepLab, are qualitatively similar. For GAP-DeepLab, we observe repeating patterns of stride that matches the overall stride of the DeepLab network - we conjecture that the pattern and the bad performance is due to the dilation-sparsified filters.

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.

9.4 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 (§9.2).

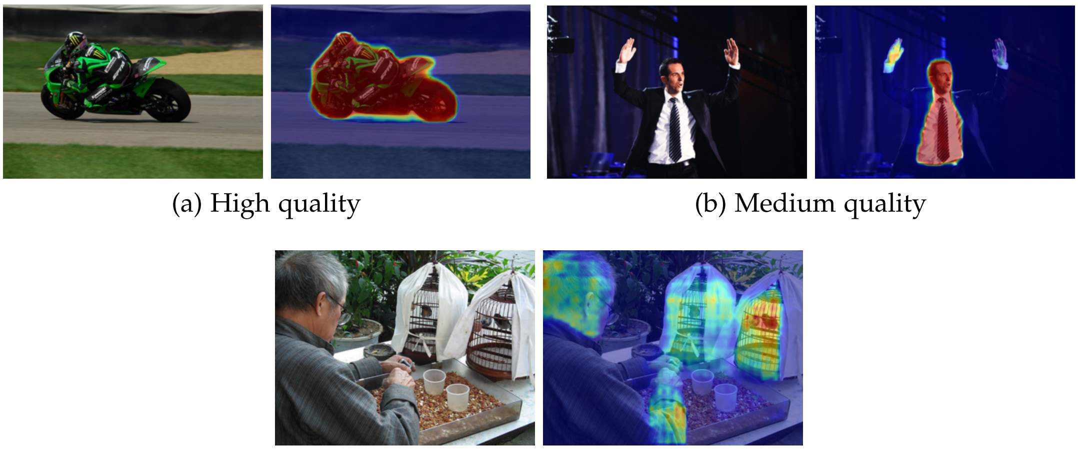
Without any prior knowledge, it is very hard, if not impossible, to learn the extent of objects only from images and image-level labels only. Image-level labels only convey information about commonly occurring patterns that are present in images with positive tags and absent in images with negative tags. 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 9.1 in previous section), yet using the same CRF setup provides +13 mIoU percent points in one, but only +7 pp on the other. In comparison our saliency approach (see below) will provide +17 pp and +18 pp for these two networks respectively.

9.4.1 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 object a person might look at first (Li *et al.*, 2014b), or a binary mask segmenting the



(a) High quality

(b) Medium quality

(c) Low quality

Figure 9.5: Example of saliency maps on Pascal images.

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.*, 2012) 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 §9.5.1) 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 9.5a, the person-bike is well segmented, but person and bike are not separated. Yet the ideal Guide labeller (Figure 9.2) should give different labels to these two objects. The second difficulty, clearly visible in the examples of Figure 9.5, is that the salient object might not belong to a category of interest (shirt instead of person in Figure 9.5b) or that the method fails to identify any salient region at all (Figure 9.5c). More random examples of our saliency model are in Figure 9.7.

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 §9.5, 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

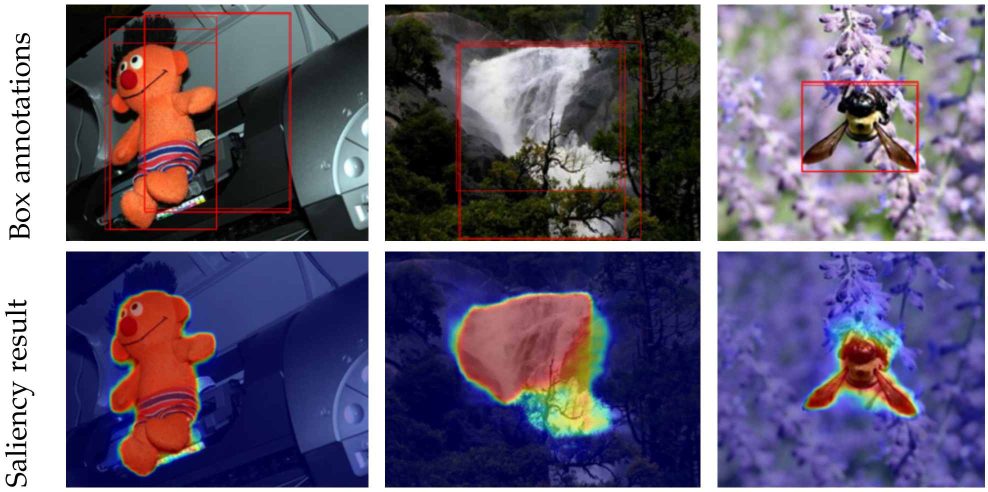


Figure 9.6: Example of saliency results on its training data. We use MSRA box annotations to train a weakly supervised saliency model. Note that the MSRA subset employed does not contain Pascal categories.

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 §9.5.1).

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 (§9.3). We want to combine them in such a way that seed signals are well propagated throughout the foreground saliency mask. 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 §9.5.1). 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.

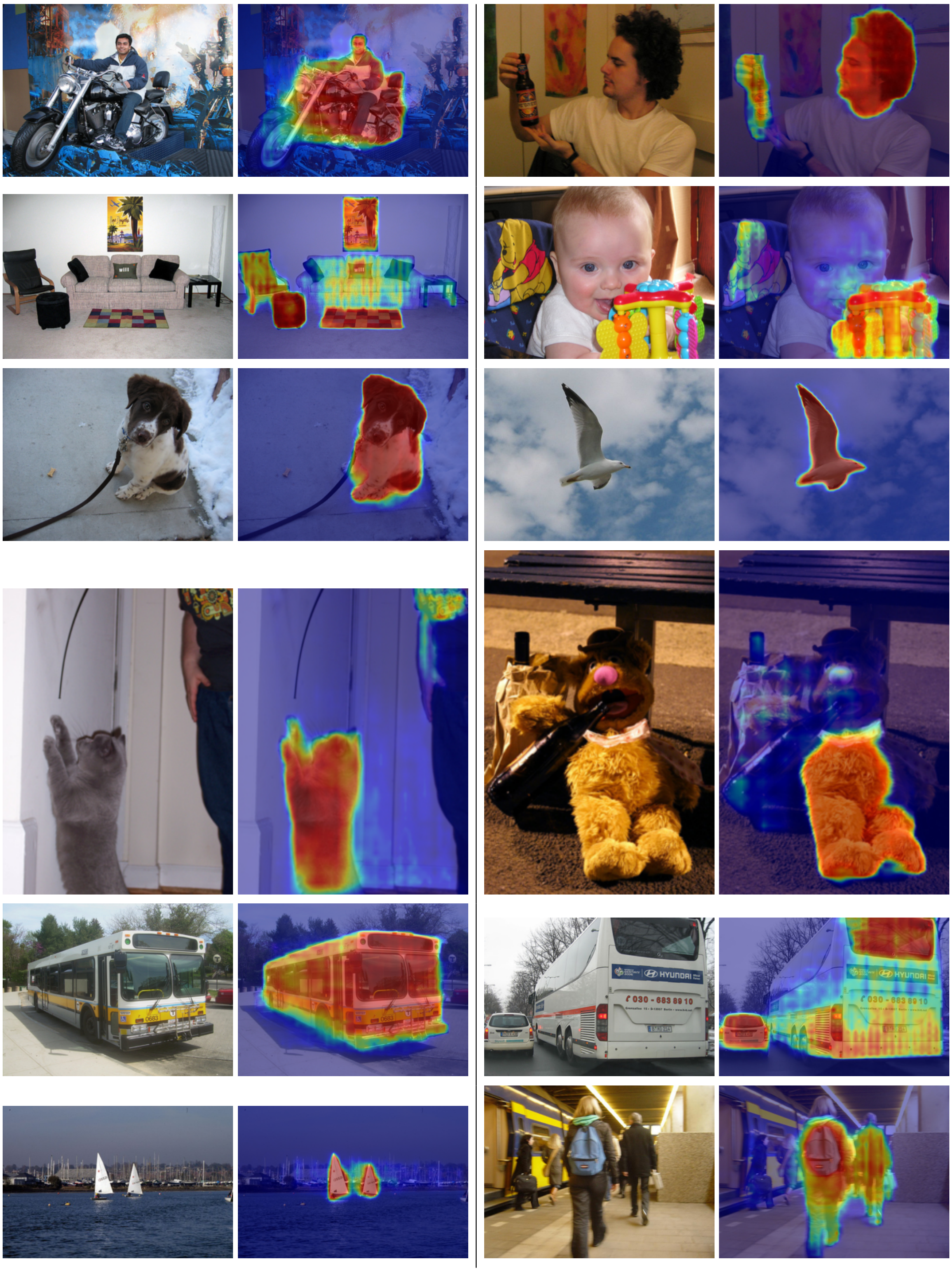


Figure 9.7: Extension of Figure 9.5. Example of saliency results on Pascal images. We note that the saliency often fails when the central, salient objects are non-Pascal or when the scene is cluttered. Examples are chosen at random.

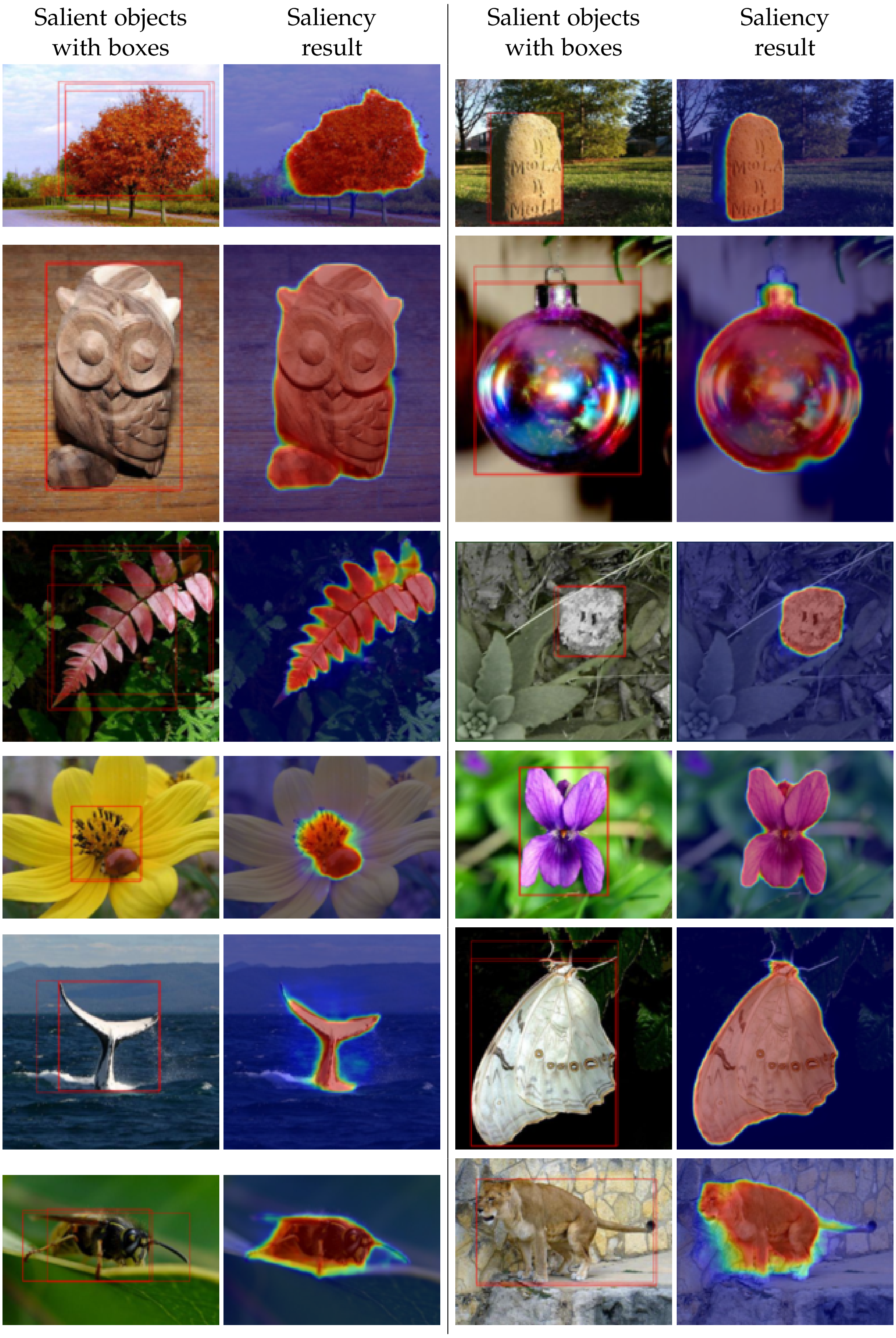
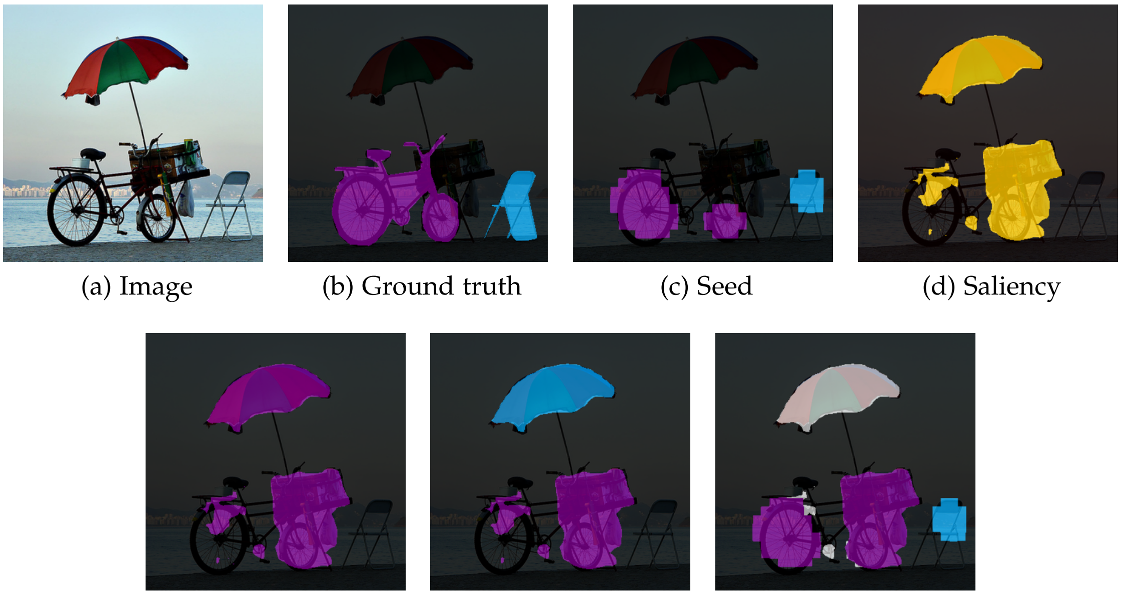


Figure 9.8: Extension of Figure 9.6. Examples of saliency results on its training data. We use MSRA box annotations to train a weakly supervised saliency model. Note that the MSRA subset employed is not biased towards the Pascal categories. Examples are chosen at random.



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

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

Figure 9.9: Guide labelling strategies example results. The image, its labels ("bicycle, chair"), seeds, and saliency map are their input. White overlay indicates "ignore" pixel label.

$\mathcal{G}_2$ **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 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 9.9 provides example results of the different guide strategies. For additional qualitative examples of seeds, saliency foreground, and generated labels, see Figure 9.10. With our guide strategies $\mathcal{G}_0$, $\mathcal{G}_1$, and $\mathcal{G}_2$ at hand, we now proceed to empirically evaluate them in §9.5.

9.5 EXPERIMENTS

We empirically evaluate our proposed method in this section. §9.5.1 provides the implementation details and the evaluation metric. §9.5.2 compares our different guide strategies amongst each other, and §9.5.3 compares with previous work on weakly supervised semantic labelling from image-level labels.

9.5.1 Implementation details and evaluation

Seeder. The final results in Tables 9.2 and 9.3 are obtained using `GAP-HighRes` (see S9.3), trained for image classification on the Pascal *trainaug* set (10 582 images), an extension of the original *train* set (1 464 images) (Everingham *et al.*, 2012; Hariharan *et al.*, 2011). This is the same procedure used by previous work on fully supervised (Chen *et al.*, 2016) and weakly supervised (Kolesnikov and Lampert, 2016b) semantic segmentation. The test time foreground threshold τ is set to 0.2, following the previous literature (Zhou *et al.*, 2016; Kolesnikov and Lampert, 2016b).

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.*, 2016) 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.

MSRA 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 (Khoreva *et al.*, 2017). 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 9.6. 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.

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.*, 2016) architecture as our segmenter convnet. The network is trained on the Pascal *trainaug* set with 10 582 images, using the output of the guide labeller (§9.2), 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 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.*, 2016).

$\mathcal{G}_1$ **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.

Method	Seeds	Saliency	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 9.2: Comparison of different guide labeller variants. Pascal VOC 2012 *val* set results, without CRF post-processing. Fg/Bg P/R: are foreground/background precision and recall of the guide labels. Discussion in §9.5.2.

Evaluation. We evaluate our image-level supervised semantic segmentation system on the Pascal VOC 2012 segmentation benchmark (Everingham *et al.*, 2012). We report all the intermediate results on the *val* 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.

9.5.2 Ingredients study

Table 9.2 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 → 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 → 51.2). This shows that the details of how a classifier is used can make a large difference.

Table 9.2 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 → 56.9). This shows that the quality of the saliency is an important ingredient, and that there is room for further gains.

Figure 9.10 contains examples of the guide labelling strategies $\mathcal{G}_0$, $\mathcal{G}_1$, and $\mathcal{G}_2$. $\mathcal{G}_0$ 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.

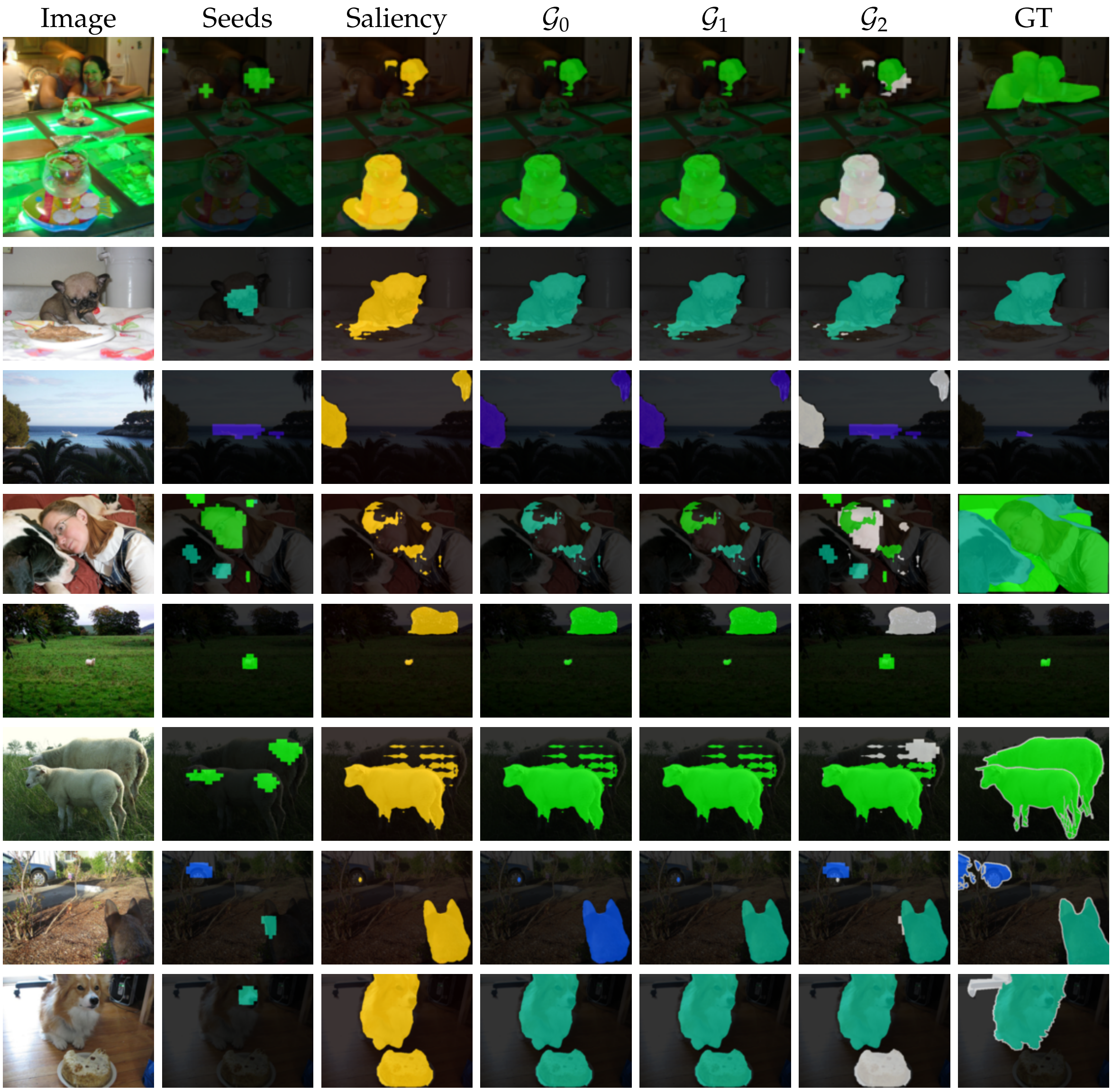


Figure 9.10: Extension of Figure 9.9. Example results for three different guide labelling strategies, $\mathcal{G}_0$, $\mathcal{G}_1$, and $\mathcal{G}_2$. The image, its image labels, seeds, and saliency map are their input. White labels indicate "ignore" regions. Examples are chosen at random.

	Method	Citation	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	Ours	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
	$\mathcal{G}_0$	Ours	I+P+S	48.8	-	-
	$\mathcal{G}_2$	Ours	I+P+S	**55.7**	**56.7**	**80.6**
	DeepLabv1	Chen *et al.* (2016)	I+P_{full}	67.6	70.3	100

Table 9.3: Comparison of state-of-the-art methods, on Pascal VOC 2012 *val* and *test* sets. 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.

9.5.3 Test set results and comparison

Table 9.3 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 legend Table 9.3), 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[23]. 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.

Some qualitative results are presented in Figure 9.11. We observe that the 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.

[23] Qi *et al.* (2016) also report 54.3 *val* set results; however, we do not consider these results comparable since they use the MCG scores (Pont-Tuset *et al.*, 2017), which are trained on the ground truth Pascal segments.

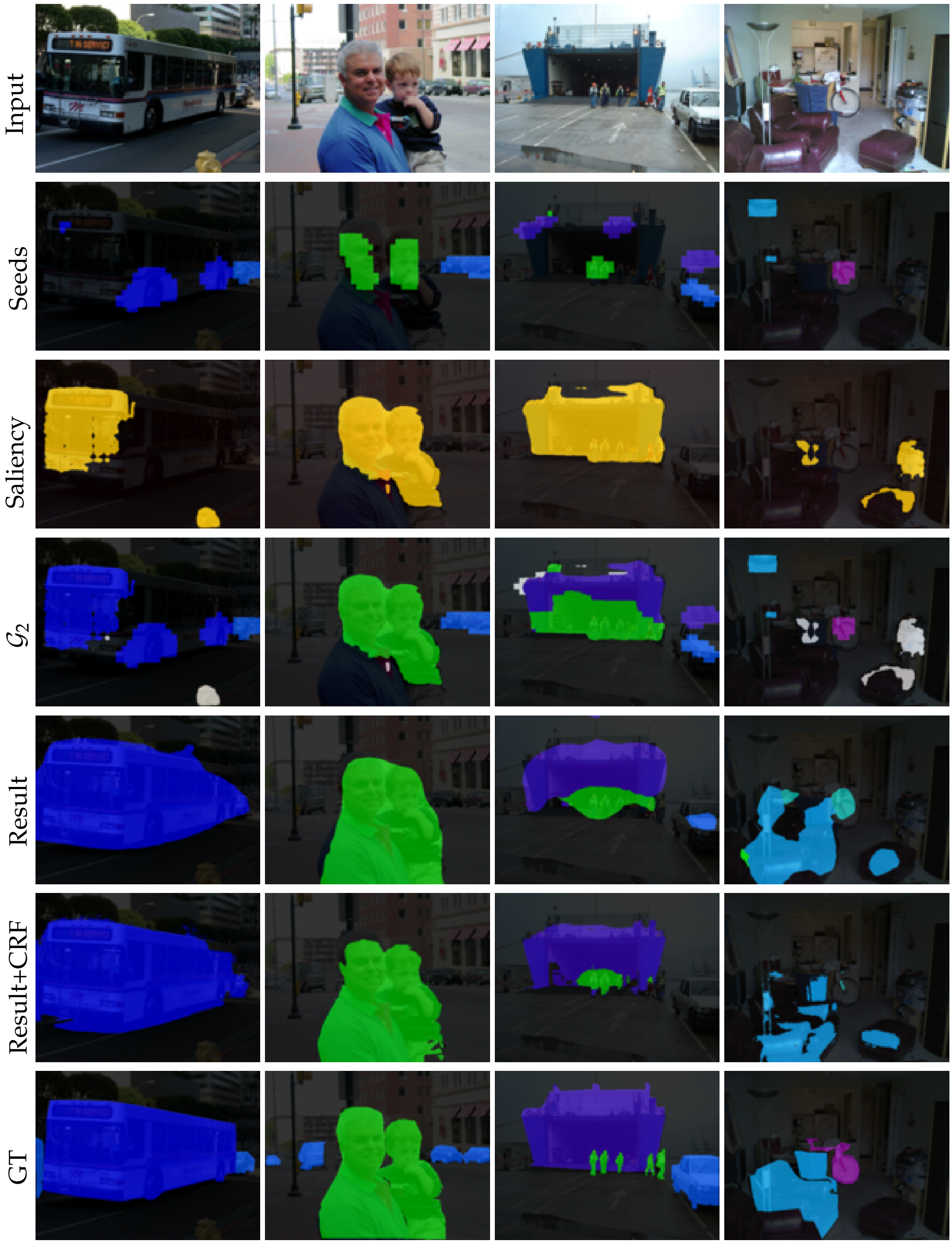


Figure 9.11: Qualitative examples of the different stages of the Guided Segmentation system on the training images. White labels are “ignore” regions.

9.6 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 (§9.2), 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.

CONCLUSION AND FUTURE PERSPECTIVES 10

We have studied the significant and timely problem of data manipulation against learned models, focusing on its implications on privacy and security relevant applications using visual data. This chapter summarises key insights and techniques developed in the previous chapters. We close the thesis with future perspectives and research directions.

10.1 KEY INSIGHTS AND CONCLUSIONS

In this section, we walk through each chapter, with a brief summary of key insights and methodologies developed with regards to the privacy and security relevant scenarios.

Data Manipulation Framework. In §1.1, we have introduced the data manipulation framework. The manipulator, characterised by the (1) goal, (2) leverage, and (3) knowledge, transforms the input for a learned model to derive a desired outcome, resulting in multiple privacy and security relevant scenarios. This framework has provided a common perspective on the following chapters.

Part I: Privacy Analysis in Visual Data. We have studied the privacy implications of data manipulation first. We have developed a state of the art person recognition system based on deep neural networks (`naeil`) and studied the identifiability of humans in personal photos, under natural domain shifts (e.g. cloth changes, Chapter 3) and intentional data degradation (e.g. face blurring, Chapter 4). Our experiments show that `naeil` is robust against such image manipulations when contexts are available (e.g. body and scene regions or photo-album metadata). We have contributed to raising alertness in public[24] and starting academic discussions in this crucial interdisciplinary area (Chapter 2).

Part II: Privacy Solution in Visual Data. To address the privacy issues thus exposed, we have presented identity obfuscation techniques to protect subjects appearing in photos. In Chapter 5, we have introduced a head inpainting technique that generates plausible faces of non-existent identities. Our method has resulted in more natural and effective identity obfuscation than face blurring or blacking-out. In Chapter 7, we have proposed an image perturbation technique that nearly perfectly obfuscates identities while modifying the data within a human-perception boundary.

[24] http://www.dailymail.co.uk/sciencetech/article-3730045/Researchers-develop-Faceless-Recognition-identify-hidden-faces-photos.html

For this performance, indeed, we have required a strong assumption on the manipulator's knowledge on the target system, namely that the full gradient computation should be accessible. Chapter 7 discusses ways to relax this assumption through game theory, whose main conclusions we discuss in the next paragraph.

Part III: Knowledge on Target Model. Knowledge matters for effective data manipulation. In Chapter 7, we have observed that while the full specification of the target model will enable effective data perturbations, an incomplete knowledge on the target (e.g. candidate space) can still guarantee certain level of effectiveness. We have employed a game theoretical analysis to obtain privacy guarantees for a user employing the adversarial example based obfuscation techniques. In Chapter 8, we discuss how one can *increase* knowledge about the target model only through a series of queries (black-box access). We have developed model exposing techniques `kennen` that learn the correlation between model internals (e.g. function class, optimisation algorithm, and training data) and the input-output patterns. We have shown that the knowledge thus obtained can craft more targeted adversarial examples. In Chapter 9, we have extracted knowledge on object locations from image classifiers to train a dense labelling network. This model has achieved 80% of the pixel-wise supervised performance with only image tag supervision.

10.2 FUTURE PERSPECTIVES

The realm of privacy and security problems in machine learning is vast. This thesis has broaden the boundary of our understanding of the problem, but there remain many unanswered questions and interesting opportunities. In this section, we point to a few potential follow-up research topics as well as more long-term future directions of the field.

10.2.1 Follow-up research topics

We introduce a few short-term research topics based on the thesis.

Gaining knowledge over time. In the data manipulation framework (§1.1), we have not considered *time*: the utility function, leverage (strategy), and knowledge level are assumed to be constant across time. In reality, knowledge is accumulated in time: through iterations of actions and observations, an agent (manipulator) learns about the environment (target model), e.g. by using `kennen` in Chapter 8). The model may also accumulate knowledge on the manipulation, if the manipulation patterns are regular. The study of equilibrium strategies for the involved agents is an interesting future research direction. In this case, the optimal solution of the agent will be an appropriate mixture of exploration (increasing knowledge), exploitation (increasing utility), and randomisation (decrease knowledge for the opponent). Optimal Control theory (Seierstad and Sydsaeter, 1986), Game theory (Nash *et al.*, 1950), and Multi-agent reinforcement learning (Tan, 1993) are relevant academic fields.

Private information other than face. While we have focused on the face and identity as crucial private information in the thesis, other they are not the only private information: political viewpoints, demographic groups, and credit card numbers can also be private. For companies, computer monitors and whiteboard notes are often considered proprietary. Users will benefit from technologies that protect various instances of private information in visual data. Orekondy *et al.* (2017) and Orekondy *et al.* (2018) have made good steps towards this direction.

More manipulation spaces. The thesis and the field in general have mostly focused on additive data manipulations. For user privacy and model security applications, "small" perturbations are preferred (§1.1). This has been enforced by some small L_p norm, as a proxy to perceptual distance for humans. However, this perturbation space excludes e.g. small translations and rotations of the image that result in big L_p distances but imperceptible changes. Such new types of manipulations are new avenues for privacy protection (Part II) and new loopholes for model security perspective (Part III). We start to see researches in this direction (Xiao *et al.*, 2018; Zeng *et al.*, 2017; Kanbak *et al.*, 2018).

Components that will improve over time. Some of our techniques will automatically improve in the future due to advances in underlying technologies and computing power. For example, ongoing advances in image generation and image inpainting (Karras *et al.*, 2018; Bora *et al.*, 2018) are likely to improve the naturalness of inpainting-based obfuscation (Chapter 5). Improvements in the performance and efficiency of general neural network architectures (He *et al.*, 2016; Huang *et al.*, 2017a) will trickle down to applications like person recogniser (Chapter 3) and semantic segmentation networks (Chapter 9).

10.2.2 Long-term perspectives

We provide long-term perspectives and discussions on machine learning, privacy, and security.

Duality of user privacy and model security. The user privacy and model security can be described as dual problems: both select a model-breaking data manipulation. This zero-sum game between the two sides leads to the question: which side will win as technology progresses? Can a learned model be free of any loophole? Or is there a "robustness ceiling" for a model bounded to finite computational resources? We do not have a good answer to those questions yet, although we see recent increase in robust machine learning papers that attempt to address those questions.

Semantics changing manipulation for user privacy. While the user privacy and model security can be posed as dual, in many applications they do not necessarily need to be. For model security, the attacks are required to not change the semantic meaning of the input. On the other hand, from the user privacy perspective, the obfuscation patterns are only required to be "pleasant" to users so that the data are still usable for personal photo albums or social networks. These requirements are neither equal nor strictly containing each other. In particular, even if a person recogniser has achieved robustness within the semantic class (person identity), the users still have room for natural yet semantics changing manipulations, such as caricaturisation (Hassan *et al.*, 2017) or head replacement (Chapter 5). Advances in model security does not necessarily imply decrease in user privacy.

Robust models. There is an active ongoing research on the robustness of learned models. Traditionally, learning theory has provided stochastic lower bounds on performance on iid (Vapnik and Chervonenkis, 2015) or domain-shifted (Ben-David *et al.*, 2010) test data. With the discovery of adversarial examples, *robust machine learning* is gaining more popularity. Instead of stochastic bounds, deterministic guarantees on the performance are obtained around training data points (Hein and Andriushchenko, 2017; Carlini *et al.*, 2018; Sinha *et al.*, 2018). While providing hard lower bounds, their results do not generalise even to iid test samples yet. Given the high dimensionality of data and parameter spaces, obtaining theoretical guarantees and devising robust training procedures are highly challenging. A breakthrough in this area, however, will be rewarding: numerous security-critical applications will benefit from the scalability of machine learning.

LIST OF FIGURES

1.1 Privacy and security framework 2
1.2 Adversarial examples 3
1.3 Obfuscation examples 4
3.1 Person recognition problem 23
3.2 PIPA split visualisation 25
3.3 PIPA detection versus annotation statistics 27
3.4 PIPA detection examples 27
3.5 Image regions for feature extraction 28
3.6 Effect of fine-tuning 29
3.7 Determining λ 36
3.8 Relative accuracy across the time gap 41
3.9 Viewpoint-wise results 42
3.10 Accuracy across viewpoints 43
3.11 Accuracy with respect to input resolution 44
3.12 Accuracy with respect to training samples per identity 45
3.13 Per-identity accuracy statistics 45
3.14 Open world setup 47
3.15 Open-world results 48
3.16 Qualitative examples 49
3.17 Qualitative examples of failure cases 49
4.1 Example of privacy scenario 52
4.2 Obfuscation types 53
4.3 Impact of head obfuscation on person recognition 57
4.4 Impact of the amount of tags on person recognition 57
4.5 Example of person matching in social media 58
4.6 Person matching performance 60
4.7 Joint person recognition performance 62
4.8 Final performance with respect to the number of tags 63
4.9 Final performance with respect to various obfuscation scenarios 64
4.10 More qualitative examples 65
5.1 Example of head obfuscation via inpainting 69
5.2 Overview of the two-stage head inpainting framework 71
5.3 Landmark detection and generation (Stage-I) 72
5.4 Conditional head image generation (Stage-II) 74
5.5 Qualitative examples of head obfuscation 79
6.1 Variety in privacy preferences under similar physical, social and image usage scenarios 87
6.2 I-Pic major components 90
6.3 I-Pic workflow 92

6.4 I-Pic secure matching protocol 95
6.5 I-Pic decision tree 99
6.6 I-Pic Intent score of images 100
6.7 Computational resource analysis 104
7.1 Game between user and recogniser 109
7.2 Detailed game between user and recogniser 110
7.3 Qualitative examples of AIPs 119
7.4 Fooling rate versus perturbation size 120
7.5 More qualitative examples - I 125
7.6 More qualitative examples - II 126
7.7 More qualitative examples - III 127
8.1 Overview of a metamodel 131
8.2 Training scheme for metamodels 134
8.3 `kennen-i` examples 135
8.4 `kennen-o` performance with respect to diverse factors 138
8.5 `kennen-io` performance with respect to diverse factors 138
8.6 t-SNE embedding of query outputs for `kennen-o` 141
8.7 t-SNE embedding of query outputs for `kennen-io` 142
8.8 Confusion matrices for `kennen-o` 143
8.9 Confusion matrices for `kennen-io` 144
8.10 Examples of AIPs 147
9.1 Teaser of our method 150
9.2 High level Guided Segmentation architecture. 154
9.3 Comparison of seed techniques - precision recall plot 155
9.4 Qualitative examples of seeds 156
9.5 Example of saliency maps on Pascal images. 159
9.6 Qualitative examples of saliency on its training data 160
9.7 Example of saliency maps on Pascal images 161
9.8 More qualitative examples of saliency on its training data 162
9.9 Qualitative examples of the guide labelling strategies 163
9.10 Extension of Figure 9.9. 166
9.11 Qualitative examples of our guide labeller 168

LIST OF TABLES

Tab. 3.1 PIPA statistics . . . 24
Tab. 3.2 PIPA split statistics . . . 24
Tab. 3.3 Impact of image regions on performance . . . 30
Tab. 3.4 Analysis of scene cues . . . 31
Tab. 3.5 Analysis of the impact of extra training data . . . 32
Tab. 3.6 List of attributes considered . . . 34
Tab. 3.7 Analysis of the impact of DeepID2+ . . . 35
Tab. 3.8 Main person recognition results . . . 38
Tab. 4.1 Privacy scenarios . . . 54
Tab. 4.2 CRF inference problem size . . . 61
Tab. 4.3 Computational time and accuracy for CRF inference . . . 61
Tab. 5.1 Final evaluation of obfuscated heads . . . 76
Tab. 6.1 Age groups of survey participants . . . 88
Tab. 6.2 Variety in privacy preferences for same person . . . 88
Tab. 6.3 Experimental dataset . . . 98
Tab. 6.4 Roles of faces captured in images . . . 102
Tab. 6.5 Privacy loss scores . . . 103
Tab. 6.6 I-Pic's projected capacity . . . 105
Tab. 7.1 Conceptual comparison of AIP methods . . . 114
Tab. 7.2 Comparison of AIP performances . . . 117
Tab. 7.3 Comparison of robustness of AIPs . . . 118
Tab. 7.4 Game payoff table . . . 121
Tab. 7.5 Selective AIP results . . . 122
Tab. 7.6 Optimal strategies against different recognisers . . . 123
Tab. 7.7 Extended tables . . . 124
Tab. 8.1 MNIST classifier attributes . . . 132
Tab. 8.2 Distribution of attributes in MNIST-NETS . . . 133
Tab. 8.3 Comparison of metamodel methods . . . 137
Tab. 8.4 Extrapolation performances of metamodels . . . 139
Tab. 8.5 Details of ImageNet classifiers . . . 146
Tab. 8.6 Transferability of adversarial examples . . . 147
Tab. 8.7 Attack performance on black-box ImageNet classifiers . . . 148
Tab. 9.1 Comparisons of seed techniques - architecture and performance 155
Tab. 9.2 Comparison of guide labeller variants . . . 165
Tab. 9.3 Comparison of weakly supervised object segmentation methods on Pascal . . . 167

BIBLIOGRAPHY

P. Aditya, R. Sen, S. J. Oh, R. Benenson, B. Bhattacharjee, P. Druschel, T. Wu, M. Fritz, and B. Schiele (2016). I-Pic: A Platform for Privacy-Compliant Image Capture. 7, 67, 83

S. Ahern, D. Eckles, N. S. Good, S. King, M. Naaman, and R. Nair (2007). Over-exposed?: privacy patterns and considerations in online and mobile photo sharing, *Proceedings of the ACM SIGCHI conference on Human factors in computing systems (SIGCHI)*. 13

E. Ahmed, M. Jones, and T. K. Marks (2015). An Improved Deep Learning Architecture for Person Re-Identification, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 11

D. Anguelov, K.-c. Lee, S. B. Gokturk, and B. Sumengen (2007). Contextual identity recognition in personal photo albums, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 11, 12

G. Ateniese, G. Felici, L. V. Mancini, A. Spognardi, A. Villani, and D. Vitali (2015). Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers, *International Journal of Security and Networks (IJSN)*, vol. 10(3), pp. 137–150. 19

A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok (2018). Synthesizing Robust Adversarial Examples, *arXiv*. 15

V. Badrinarayanan, A. Kendall, and R. Cipolla (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 39(12), pp. 2481–2495. 153

S. Baluja and I. Fischer (2018). Learning to Attack: Adversarial Transformation Networks. 15

A. Bansal, X. Chen, B. Russell, A. Gupta, and D. Ramanan (2016). PixelNet: Towards a general pixel-level architecture, *arXiv*. 153

M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar (2006). Can Machine Learning Be Secure?, *Proceedings of the ACM Symposium on Information, computer and communications security (ASIACCS)*. 15

A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei (2016). What's the point: Semantic segmentation with point supervision, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 151

A. Bedagkar-Gala and S. K. Shah (2014). A survey of approaches and trends in person re-identification, *Image and Vision Computing (IVC)*. 12

S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan (2010). A theory of learning from different domains, *Machine learning*. 10, 174

A. Ben-Tal, L. El Ghaoui, and A. Nemirovski (2009). *Robust Optimization*, Princeton University Press. 5, 6, 18

A. Besmer and H. Richter Lipford (2010). Moving beyond untagging: photo privacy in a tagged world, *Proceedings of the ACM SIGCHI conference on Human factors in computing systems (SIGCHI)*. 13

B. Biggio, G. Fumera, and F. Roli (2008). Adversarial Pattern Classification Using Multiple Classifiers and Randomisation, *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. 15

B. Biggio, B. Nelson, and P. Laskov (2012). Poisoning Attacks Against Support Vector Machines, *Proceedings of the International Conference on Machine Learning (ICML)*. 17

H. Bilen and A. Vedaldi (2016). Weakly Supervised Deep Detection Networks, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 152

C. Bo, G. Shen, J. Liu, X.-Y. Li, Y. Zhang, and F. Zhao (2014). Privacy.Tag: Privacy Concern Expressed and Respected, *ACM Conference on Embedded Network Sensor Systems*. 84, 85, 88

M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.* (2016). End to end learning for self-driving cars, *arXiv*. 1

A. Bora, E. Price, and A. G. Dimakis (2018). AmbientGAN: Generative models from lossy measurements, *International Conference on Learning Representations (ICLR)*. 173

A. Borji, M.-M. Cheng, H. Jiang, and J. Li (2015). Salient Object Detection: A Benchmark, *IEEE Transactions on Image Processing (TIP)*, vol. 24(12), pp. 5706–5722. 152, 159

L. Bourdev, S. Maji, and J. Malik (2011). Describing People: Poselet-Based Attribute Classification, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 86

L. Bourdev and J. Malik (2009). Poselets: Body part detectors trained using 3d human pose annotations, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 12, 38

K. Brkic, I. Sikiric, T. Hrkac, and Z. Kalafatic (2017). I Know That Person: Generative Full Body and Face De-identification of People in Images, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 14

M. Brückner, C. Kanzow, and T. Scheffer (2012). Static Prediction Games for Adversarial Learning Problems, *Journal of Machine Learning Research (JMLR)*, vol. 13(Sep), pp. 2617–2654. 18, 113

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, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 151

N. Carlini, G. Katz, C. Barrett, and D. L. Dill (2018). Ground-Truth Adversarial Examples, *arXiv*. 17, 174

N. Carlini and D. Wagner (2016). Defensive distillation is not robust to adversarial examples, *arXiv*. 16

N. Carlini and D. Wagner (2017a). Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods, *Proceedings of the ACM Workshop on Artificial Intelligence and Security (AISEC)*. 16

N. Carlini and D. Wagner (2017b). Towards Evaluating the Robustness of Neural Networks, *IEEE Symposium on Security and Privacy (SP)*. 15

B.-C. Chen, C.-S. Chen, and W. H. Hsu (2014). Cross-Age Reference Coding for Age-Invariant Face Recognition and Retrieval, *Proceedings of the European Conference on Computer Vision (ECCV)*. 32, 38

D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun (2012). Bayesian Face Revisited: A Joint Formulation, *Proceedings of the European Conference on Computer Vision (ECCV)*. 34

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille (2016). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, *arXiv*. 153, 154, 164, 167

P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh (2017a). ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks Without Training Substitute Models, *Proceedings of the ACM Workshop on Artificial Intelligence and Security (AISEC)*. 19

W. Chen, X. Chen, J. Zhang, and K. Huang (2017b). Beyond triplet loss: a deep quadruplet network for person re-identification, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 11

D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng (2016). Person Re-Identification by Multi-Channel Parts-Based CNN With Improved Triplet Loss Function, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 11

D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino (2011). Custom Pictorial Structures for Re-identification, *Proceedings of the British Machine Vision Conference (BMVC)*. 11

M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu (2015). Global Contrast based Salient Region Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 37(3), pp. 569–582. 153

F. Cole, D. Belanger, D. Krishnan, A. Sarna, I. Mosseri, and W. T. Freeman (2017). Synthesizing Normalized Faces from Facial Identity Features, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 69, 70

T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham (1995). Active shape models-their training and application, *Computer Vision and Image Understanding (CVIU)*, vol. 61(1), pp. 38–59. 72

G. V. Cormack (2008). Email Spam Filtering: A Systematic Review, *Foundations and Trends in Information Retrieval*. 15

J. Cui, F. Wen, R. Xiao, Y. Tian, and X. Tang (2007). EasyAlbum: an interactive photo annotation system based on face clustering and re-ranking, *Proceedings of the ACM SIGCHI conference on Human factors in computing systems (SIGCHI)*. 12

J. Dai, K. He, and J. Sun (2015). Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 151

L. D'Antoni, A. Dunn, S. Jana, T. Kohno, B. Livshits, D. Molnar, A. Moshchuk, E. Ofek, F. Roesner, S. Saponas, M. Veanes, and H. J. Wang (2013). Operating System Support for Augmented Reality Applications, *HotOS Workshop*. 85

T. Darrell, M. Kloft, M. Pontil, G. Rätsch, and E. Rodner (2015). Machine Learning with Interdependent and Non-identically Distributed Data (Dagstuhl Seminar 15152), *Dagstuhl Reports*. 9

N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, L. Chen, M. E. Kounavis, and D. H. Chau (2017). Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression, *arXiv*. 16

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). Imagenet: A large-scale hierarchical image database, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 9, 12, 16, 38, 108

Y. Deng, P. Luo, C. C. Loy, and X. Tang (2014). Pedestrian attribute recognition at far distance, *Proceedings of the ACM international conference on Multimedia (ACMMM)*. 33, 38

T. Denning, Z. Dehlawi, and T. Kohno (2014). In Situ with Bystanders of Augmented Reality Glasses: Perspectives on Recording and Privacy-Mediating Technologies, *Proceedings of the ACM SIGCHI conference on Human factors in computing systems (SIGCHI)*. 85

G. S. Dhillon, K. Azizzadenesheli, J. D. Bernstein, J. Kossaifi, A. Khanna, Z. C. Lipton, and A. Anandkumar (2018). Stochastic activation pruning for robust adversarial defense, *International Conference on Learning Representations (ICLR)*. 18

X. Di, V. A. Sindagi, and V. M. Patel (2017). GP-GAN: Gender Preserving GAN for Synthesizing Faces from Landmarks, *arXiv*. 70

G. K. Dziugaite, Z. Ghahramani, and D. M. Roy (2016). A study of the effect of jpg compression on adversarial images, *arXiv*. 16

K. Ehsani, R. Mottaghi, and A. Farhadi (2018). SeGAN: Segmenting and Generating the Invisible, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 70

M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman (2015). The pascal visual object classes challenge: A retrospective, *International Journal on Computer Vision (IJCV)*, vol. 111(1), pp. 98–136. 9

M. Everingham, J. Sivic, and A. Zisserman (2006). Hello! My name is... Buffy–automatic naming of characters in TV video, *Proceedings of the British Machine Vision Conference (BMVC)*. 12

M. Everingham, J. Sivic, and A. Zisserman (2009). Taking the bite out of automated naming of characters in TV video, *Image and Vision Computing (IVC)*. 12

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman (2012). The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. 155, 159, 164, 165

A. Gallagher and T. Chen (2008). Clothing Cosegmentation for Recognizing People, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 11, 12, 23, 54

A. C. Gallagher and T. Chen (2007). Using group prior to identify people in consumer images, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 56

Y. Ganin and V. Lempitsky (2015). Unsupervised Domain Adaptation by Backpropagation, *Proceedings of the International Conference on Machine Learning (ICML)*. 10

Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky (2016). Domain-adversarial training of neural networks, *Journal of Machine Learning Research (JMLR)*, vol. 17(1), pp. 2096–2030. 10

B. Goethals, S. Laur, H. Lipmaa, and T. Mielikainen (2004). On private scalar product computation for privacy-preserving data mining, *International Conference on Information Security and Cryptology (ICISC)*. 90, 94

S. Gong, M. Cristani, S. Yan, and C. C. Loy (2014). *Person re-identification*, Springer. 11

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio (2014). Generative Adversarial Nets, *Advances in Neural Information Processing Systems (NIPS)*. 67

I. J. Goodfellow, J. Shlens, and C. Szegedy (2015). Explaining and harnessing adversarial examples, *International Conference on Learning Representations (ICLR)*. 5, 15, 16, 108, 112, 114, 146

R. Gopalan, S. Taheri, P. Turaga, and R. Chellappa (2012). A blur-robust descriptor with applications to face recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 34(6), pp. 1220–1226. 13

A. Graese, A. Rozsa, and T. E. Boult (2016). Assessing Threat of Adversarial Examples on Deep Neural Networks, *IEEE International Conference on Machine Learning and Applications (ICMLA)*. 15, 109, 112, 118

D. Gray, S. Brennan, and H. Tao (2007). Evaluating appearance models for recognition, reacquisition, and tracking, *IEEE International Workshop on Performance Evaluation for Tracking and Surveillance (PETS)*. 11

K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. D. McDaniel (2017). On the (Statistical) Detection of Adversarial Examples, *arXiv*. 16

S. Gupta, J. Hoffman, and J. Malik (2016). Cross Modal Distillation for Supervision Transfer, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 72

D. Hall and P. Perona (2015). Fine-Grained Classification of Pedestrians in Video: Benchmark and State of the Art, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 11

P. Hallgren, M. Ochoa, and A. Sabelfeld (2015). InnerCircle: A Parallelizable Decentralized Privacy-Preserving Location Proximity Protocol, *IEEE Annual Conference on Privacy, Security and Trust (PST)*. 86

L. Hansen and T. J. Sargent (2001). Robust control and model uncertainty, *American Economic Review*. 5, 18

B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik (2011). Semantic Contours from Inverse Detectors, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 157, 164

E. T. Hassan, R. Hasan, P. Shaffer, D. J. Crandall, and A. Kapadia (2017). Cartooning for Enhanced Privacy in Lifelogging and Streaming Videos, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 14, 174

Z. Hayder, X. He, and M. Salzmann (2015). Structural kernel learning for large scale multiclass object co-detection, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 56

J. Hayes and G. Danezis (2017). Machine Learning as an Adversarial Service: Learning Black-Box Adversarial Examples, *arXiv*. 20

K. He, X. Zhang, S. Ren, and J. Sun (2016). Deep Residual Learning for Image Recognition, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 9, 38, 39, 108, 116, 146, 153, 173

M. Hein and M. Andriushchenko (2017). Formal Guarantees on the Robustness of a Classifier against Adversarial Manipulation, *Advances in Neural Information Processing Systems (NIPS)*. 17, 174

J. Holvast (2008). History of privacy, *IFIP Summer School on the Future of Identity in the Information Society*. 13

R. Hoyle, R. Templeman, S. Armes, D. Anthony, D. Crandall, and A. Kapadia (2014). Privacy Behaviors of Lifeloggers using Wearable Cameras, *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*. 85

Y. Hu, D. Yi, S. Liao, Z. Lei, and S. Li (2014). Cross Dataset Person Re-identification, *Proceedings of the Asian Conference on Computer Vision (ACCV)*. 11

G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger (2017a). Densely connected convolutional networks, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 146, 173

G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller (2007). Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, Technical report, UMass. 9, 11, 12, 22, 34, 93

R. Huang, B. Xu, D. Schuurmans, and C. Szepesvári (2015). Learning with a Strong Adversary, *arXiv*. 16, 112

S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel (2017b). Adversarial attacks on neural network policies, *arXiv*. 15

Y. Huang, L. Malka, D. Evans, and J. Katz (2011). Efficient Privacy-Preserving Biometric Identification, *NDSS*. 94

F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size, *arXiv*. 145

S. Ioffe and C. Szegedy (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, *Proceedings of the International Conference on Machine Learning (ICML)*. 146

Y. Ishai, J. Kilian, K. Nissim, and E. Petrank (2003). Extending Oblivious Transfers Efficiently, *Annual International Cryptology Conference (AICC)*. 96

S. Jana, D. Molnar, A. Moshchuk, A. Dunn, B. Livshits, H. J. Wang, and E. Ofek (2013a). Enabling Fine-Grained Permissions for Augmented Reality Applications With Recognizers, *Usenix Security*. 85

S. Jana, A. Narayanan, and V. Shmatikov (2013b). A Scanner Darkly: Protecting User Privacy from Perceptual Applications, *SP*. 85

H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li (2013). Salient object detection: A discriminative regional feature integration approach, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 153, 159

J. Johnson, A. Alahi, and L. Fei-Fei (2016). Perceptual losses for real-time style transfer and super-resolution, *Proceedings of the European Conference on Computer Vision (ECCV)*. 1

J. Jung and M. Philipose (2014). Courteous Glass, *UPSIDE, Ubicomp Workshop*. 85

C. Kanbak, S.-M. Moosavi-Dezfooli, and P. Frossard (2018). Geometric robustness of deep networks: analysis and improvement, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 15, 173

V. Kantorov, M. Oquab, M. Cho, and I. Laptev (2016). ContextLocNet: Context-Aware Deep Network Models for Weakly Supervised Localization, *Proceedings of the European Conference on Computer Vision (ECCV)*. 152

T. Karras, T. Aila, S. Laine, and J. Lehtinen (2018). Progressive growing of gans for improved quality, stability, and variation, *International Conference on Learning Representations (ICLR)*. 173

V. Kazemi and J. Sullivan (2014). One Millisecond Face Alignment with an Ensemble of Regression Trees, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 71, 75

I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard (2016). The megaface benchmark: 1 million faces for recognition at scale, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 11, 12

A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele (2017). Weakly Supervised Semantic Labelling and Instance Segmentation, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 151, 164

B. F. Klare, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain (2015). Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 11

P. W. Koh and P. Liang (2017). Understanding Black-box Predictions via Influence Functions, *Proceedings of the International Conference on Machine Learning (ICML)*. 17

A. Kolesnikov and C. Lampert (2016a). Improving Weakly-Supervised Object Localization by Micro-Annotation, *Proceedings of the British Machine Vision Conference (BMVC)*. 150, 152, 167

A. Kolesnikov and C. H. Lampert (2016b). Seed, Expand and Constrain: Three Principles for Weakly-Supervised Image Segmentation, *Proceedings of the European Conference on Computer Vision (ECCV)*. 151, 152, 154, 158, 164, 167

J. Z. Kolter and E. Wong (2017). Provable defenses against adversarial examples via the convex outer adversarial polytope, *arXiv*. 18

P. Krähenbühl and V. Koltun (2011). Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. 157, 158, 164

A. Krizhevsky, I. Sutskever, and G. E. Hinton (2012). ImageNet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems (NIPS)*. 9, 12, 15, 28, 38, 108, 116

V. Kumar, A. Namboodiri, M. Paluri, and C. Jawahar (2017). Pose-Aware Person Recognition, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 12, 38, 39

A. Kurakin, I. Goodfellow, and S. Bengio (2017a). Adversarial examples in the physical world, *International Conference on Learning Representations Workshop (ICLRW)*. 15, 114, 115

A. Kurakin, I. J. Goodfellow, and S. Bengio (2017b). Adversarial Machine Learning at Scale, *International Conference on Learning Representations (ICLR)*. 16

J. D. Lafferty, A. McCallum, and F. C. N. Pereira (2001). Conditional random fields: probabilistic models for segmenting and labeling sequence data, *Proceedings of the International Conference on Machine Learning (ICML)*. 158

G. R. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan (2003). A Robust Minimax Approach to Classification. 14, 18

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner (1998). Gradient-based learning applied to document recognition. 131

G. Li and Y. Yu (2016). Deep Contrast Learning for Salient Object Detection, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 153, 164

H. Li, J. Brandt, Z. Lin, X. Shen, and G. Hua (2016a). A Multi-Level Contextual Model For Person Recognition in Photo Albums, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 12, 38, 40

W. Li, R. Zhao, and X. Wang (2012). Human reidentification with transferred metric learning, *Proceedings of the Asian Conference on Computer Vision (ACCV)*. 11

W. Li, R. Zhao, T. Xiao, and X. Wang (2014a). DeepReID: Deep filter pairing neural network for person re-identification, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 11

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 Transactions on Image Processing (TIP)*, vol. 25(8), pp. 3919–3930. 153, 164

Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille (2014b). The secrets of salient object segmentation, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 152, 158

Y. Li, G. Lin, B. Zhuang, L. Liu, C. Shen, and A. v. d. Hengel (2017). Sequential Person Recognition in Photo Albums with a Recurrent Network, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 12

D. Lin, J. Dai, J. Jia, K. He, and J. Sun (2016). ScribbleSup: Scribble-Supervised Convolutional Networks for Semantic Segmentation, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 151

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, *Proceedings of the European Conference on Computer Vision (ECCV)*. 26, 159

Y. Lindell (2013). Fast Cut-and-Choose Based Protocols for Malicious and Covert Adversaries, *Journal of Cryptology*. 97

Z. Lingli and L. Jianghuang (2010). Security Algorithm of Face Recognition Based on Local Binary Pattern and Random Projection, *IEEE International Conference on Cognitive Informatics (ICCI)*. 86

J.-L. Lisani, A.-B. Petro, and C. Sbert (2012). Color and Contrast Enhancement by Controlled Piecewise Affine Histogram Equalization, *SIAM journal on imaging sciences*. 97

T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum (2011). Learning to detect a salient object, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 33(2), pp. 353–367. 164

Y. Liu, X. Chen, C. Liu, and D. X. Song (2017a). Delving into Transferable Adversarial Examples and Black-box Attacks, *International Conference on Learning Representations (ICLR)*. 15, 20, 113, 115, 147

Y. Liu, H. Li, and X. Wang (2017b). Rethinking Feature Discrimination and Polymerization for Large-scale Recognition, *arXiv*. 12, 38, 39, 40

J. Long, E. Shelhamer, and T. Darrell (2015a). Fully Convolutional Networks for Semantic Segmentation, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 153

M. Long, Y. Cao, J. Wang, and M. I. Jordan (2015b). Learning Transferable Features with Deep Adaptation Networks, *Proceedings of the International Conference on Machine Learning (ICML)*. 10

D. Lowd and C. Meek (2005). Adversarial Learning, *The International Conference on Mobile Systems, Applications, and Services (MobiSys)*. 15

J. Lu, T. Issaranon, and D. Forsyth (2017a). Safetynet: Detecting and rejecting adversarial examples robustly, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 16

J. Lu, H. Sibai, E. Fabry, and D. A. Forsyth (2017b). NO Need to Worry about Adversarial Examples in Object Detection in Autonomous Vehicles, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 15

Z. Lu, Z. Li, J. Cao, R. He, and Z. Sun (2017c). Recent Progress of Face Image Synthesis, *arXiv*. 69

L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool (2017). Pose guided person image generation, *Advances in Neural Information Processing Systems (NIPS)*. 70, 77

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu (2018). Towards Deep Learning Models Resistant to Adversarial Attacks, *International Conference on Learning Representations (ICLR)*. 16, 18

R. Margolin, A. Tal, and L. Zelnik-Manor (2013). What makes a patch distinct?, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 153

C. S. Mathialagan, A. C. Gallagher, and D. Batra (2015). VIP: Finding Important People in Images, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 12

M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool (2014). Face detection without bells and whistles, *Proceedings of the European Conference on Computer Vision (ECCV)*. 26, 31, 89, 93, 97

R. McPherson, R. Shokri, and V. Shmatikov (2016). Defeating Image Obfuscation with Deep Learning, *arXiv*. 13, 108

J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff (2017). On Detecting Adversarial Perturbations, *International Conference on Learning Representations (ICLR)*. 16

A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel (2010). You are who you know: inferring user profiles in online social networks, *Proceedings of the ACM international conference on Web search and data mining (WSDM)*. 13

S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard (2017). Universal Adversarial Perturbations, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 15, 20, 108, 112, 114

S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard (2016). Deepfool: a simple and accurate method to fool deep neural networks, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 15, 108, 114

S.-M. Moosavi-Dezfooli, A. Shrivastava, and O. Tuzel (2018). Divide, Denoise, and Defend against Adversarial Attacks, *arXiv*. 16

A. C. Müller and S. Behnke (2014). pystruct - Learning Structured Prediction in Python, *Journal of Machine Learning Research (JMLR)*, vol. 15(1), pp. 2055–2060. 61

M. Naor and B. Pinkas (2005). Computationally Secure Oblivious Transfer, *Journal of Cryptology*. 96

A. Narayanan and V. Shmatikov (2009). De-anonymizing social networks, *IEEE Symposium on Security and Privacy (SP)*. 13

A. Narayanan and V. Shmatikov (2010). Myths and fallacies of personally identifiable information, *Communications of the ACM (CACM)*. 13, 51, 66

N. Narodytska and S. P. Kasiviswanathan (2017). Simple Black-Box Adversarial Perturbations for Deep Networks, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 19, 113

J. F. Nash *et al.* (1950). Equilibrium points in n-person games, *Proceedings of the national academy of sciences*. 5, 18, 113, 172

A. Nech and I. Kemelmacher-Shlizerman (2017). Level Playing Field For Million Scale Face Recognition, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 11

J. v. Neumann (1928). Zur Theorie der Gesellschaftsspiele, *Mathematische Annalen*. 110, 111

H. Noh, S. Hong, and B. Han (2015). Learning deconvolution network for semantic segmentation, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 153

S. J. Oh, M. Augustin, B. Schiele, and M. Fritz (2018). Towards Reverse-Engineering Black-Box Neural Networks, *International Conference on Learning Representations (ICLR)*. 3, 7, 19, 20, 107, 129

S. J. Oh, R. Benenson, M. Fritz, and B. Schiele (2015). Person Recognition in Personal Photo Collections, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 7, 12, 21, 22, 33, 38, 39, 40, 54, 108

S. J. Oh, R. Benenson, M. Fritz, and B. Schiele (2016). Faceless Person Recognition; Privacy Implications in Social Media, *Proceedings of the European Conference on Computer Vision (ECCV)*. 7, 12, 13, 21, 51, 69

S. J. Oh, R. Benenson, M. Fritz, and B. Schiele (2017a). Person Recognition in Social Media Photos, *arXiv*. 7, 21, 22, 38

S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, and B. Schiele (2017b). Exploiting Saliency for Object Segmentation from Image Level Labels, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 8, 107, 149

S. J. Oh, M. Fritz, and B. Schiele (2017c). Adversarial Image Perturbation for Privacy Protection – A Game Theory Perspective, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 6, 7, 14, 15, 67, 68, 107, 108

A. Oliva and A. Torralba (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope, *International Journal on Computer Vision (IJCV)*, vol. 42(3), pp. 145–175. 30

T. Orekondy, M. Fritz, and B. Schiele (2018). Connecting Pixels to Privacy and Utility: Automatic Redaction of Private Information in Images, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 173

T. Orekondy, B. Schiele, and M. Fritz (2017). Towards a visual privacy advisor: Understanding and predicting privacy risks in images, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 13, 173

P. Paillier (1999). Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, *International Conference on the Theory and Applications of Cryptographic Techniques (TACT)*. 94

G. Papandreou, L. Chen, K. Murphy, , and A. L. Yuille (2015). Weakly- and Semi-Supervised Learning of a DCNN for Semantic Image Segmentation, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 151, 152, 167

N. Papernot, P. McDaniel, and I. Goodfellow (2016a). Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples, *arXiv*. 20, 113

N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami (2017). Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples, *Proceedings of the ACM on Conference on Computer and Communications Security (CCS)*. 19, 20, 113

N. Papernot, P. McDaniel, A. Sinha, and M. Wellman (2018). Towards the science of security and privacy in machine learning, *IEEE Symposium on Security and Privacy (SP)*. 14

N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami (2016b). Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks, *IEEE Symposium on Security and Privacy (SP)*. 16

O. M. Parkhi, A. Vedaldi, and A. Zisserman (2015). Deep Face Recognition, *Proceedings of the British Machine Vision Conference (BMVC)*. 38

D. Pathak, P. Kraehenbuehl, and T. Darrell (2015a). Constrained Convolutional Neural Networks for Weakly Supervised Segmentation, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 150, 151, 152, 167

D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros (2016). Context Encoders: Feature Learning by Inpainting, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 70

D. Pathak, E. Shelhamer, J. Long, and T. Darrell (2015b). Fully Convolutional Multi-Class Multiple Instance Learning, *International Conference on Learning Representations Workshop (ICLRW)*. 151, 167

P. Pinheiro and R. Collobert (2015). From Image-level to Pixel-level Labeling with Convolutional Network, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 150, 151, 152, 167

P. O. Pinheiro and R. Collobert (2014). Recurrent Convolutional Neural Networks for Scene Labeling, *Proceedings of the International Conference on Machine Learning (ICML)*. 153

J. Pont-Tuset, P. Arbeláez, J. Barron, F. Marques, and J. Malik (2017). Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 39(1), pp. 128–140. 167

A. Prékopa (1995), Kluwer Academic Publishers Group, Dordrecht. 5, 6, 18

A. Punnappurath, A. N. Rajagopalan, S. Taheri, R. Chellappa, and G. Seetharaman (2015). Face recognition across non-uniform motion blur, illumination, and pose, *IEEE Transactions on Image Processing (TIP)*, vol. 24(7), pp. 2067–2082. 13

X. Qi, Z. Liu, J. Shi, H. Zhao, and J. Jia (2016). Augmented Feedback in Semantic Segmentation Under Image Level Supervision, *Proceedings of the European Conference on Computer Vision (ECCV)*. 152, 167

A. Radford, L. Metz, and S. Chintala (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, *arXiv*. 73, 74

N. Raval, A. Machanavajjhala, and L. P. Cox (2017). Protecting Visual Secrets Using Adversarial Nets, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 18

N. Raval, A. Srivastava, K. Lebeck, L. P. Cox, and A. Machanavajjhala (2014). MarkIt: Privacy Markers for Protecting Visual Secrets, *UPSIDE, Ubicomp Workshop*. 87

N. Raval, A. Srivastava, A. Razeen, K. Lebeck, A. Machanavajjhala, and L. P. Cox (2016). What You Mark is What Apps See. 84, 85

S. Ren, K. He, R. Girshick, and J. Sun (2015). Faster R-CNN: Towards real-time object detection with region proposal networks, *Advances in Neural Information Processing Systems (NIPS)*. 1

F. Roesner, D. Molnar, A. Moshchuk, T. Kohno, and H. J. Wang (2014). World-Driven Access Control for Continuous Sensing, *Proceedings of the ACM on Conference on Computer and Communications Security (CCS)*. 84, 85, 87

O. Ronneberger, P. Fischer, and T. Brox (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation, *International Conference on Medical image computing and computer-assisted intervention (MICCAI)*. 73

A. Rozsa, E. M. Rudd, and T. E. Boult (2016). Adversarial Diversity and Hard Positive Generation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 15, 114

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li (2015). ImageNet Large Scale Visual Recognition Challenge, *International Journal on Computer Vision (IJCV)*, vol. 115(3), pp. 211–252. 72

K. Saenko, B. Kulis, M. Fritz, and T. Darrell (2010). Adapting Visual Category Models to New Domains, *Proceedings of the European Conference on Computer Vision (ECCV)*. 10

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, *Proceedings of the European Conference on Computer Vision (ECCV)*. 150, 152, 167

F. Schroff, D. Kalenichenko, and J. Philbin (2015). Facenet: A unified embedding for face recognition and clustering, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1, 11, 12, 66

A. Seierstad and K. Sydsaeter (1986). *Optimal control theory with economic applications*, Elsevier North-Holland, Inc. 5, 18, 172

R. G. J. S. Shaoqing Ren, Kaiming He (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *Advances in Neural Information Processing Systems (NIPS)*. 38, 39

M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter (2016). Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition, *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (SIGSAC)*. 14, 15, 68, 115

I. Shcherbatyi and B. Andres (2016). Convexification of Learning from Constraints, *Proceedings of the German Conference on Pattern Recognition (GCPR)*. 149

J. Shi, Q. Yan, L. Xu, and J. Jia (2016). Hierarchical image saliency detection on extended cssd, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 38(4), pp. 717–729. 152, 159

W. Shimoda and K. Yanai (2016). Distinct class-specific saliency maps for weakly supervised semantic segmentation, *Proceedings of the European Conference on Computer Vision (ECCV)*. 151, 152, 167

R. Shokri, M. Stronati, C. Song, and V. Shmatikov (2017). Membership Inference Attacks Against Machine Learning Models, *IEEE Symposium on Security and Privacy (SP)*. 19

K. Simonyan, A. Vedaldi, and A. Zisserman (2014). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, *International Conference on Learning Representations Workshop (ICLRW)*. 80, 151, 154, 157

K. Simonyan and A. Zisserman (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition, *International Conference on Learning Representations (ICLR)*. 116, 146, 153, 154, 157, 164

A. Sinha, H. Namkoong, and J. Duchi (2018). Certifiable Distributional Robustness with Principled Adversarial Training, *International Conference on Learning Representations (ICLR)*. 17, 174

C. Smowton, J. R. Lorch, D. Molnar, S. Saroiu, and A. Wolman (2014). Zero-Effort Payments: Design, Deployment, and Lessons, *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*. 85

J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller (2015). Striving for Simplicity: The All Convolutional Net, *International Conference on Learning Representations Workshop (ICLRW)*. 151, 152, 154, 157

J. Steinhardt, P. W. W. Koh, and P. S. Liang (2017). Certified Defenses for Data Poisoning Attacks, *Advances in Neural Information Processing Systems (NIPS)*. 17

Z. Stone, T. Zickler, and T. Darrell (2008). Autotagging facebook: Social network context improves photo annotation, *CVPRW*. 56

Q. Sun, M. Fritz, and B. Schiele (2017). A Domain Based Approach to Social Relation Recognition, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 108

Q. Sun, L. Ma, S. J. Oh, L. van Gool, B. Schiele, and M. Fritz (2018). Natural and Effective Obfuscation by Head Inpainting, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7, 14, 67, 68

Y. Sun, X. Wang, and X. Tang (2014). Deep learning face representation from predicting 10,000 classes, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 34, 38, 40

Y. Sun, X. Wang, and X. Tang (2015). Deeply learned face representations are sparse, selective, and robust, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 11, 12, 22, 28, 34, 38, 47, 48, 66

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich (2015). Going deeper with convolutions, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 116

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna (2016). Rethinking the inception architecture for computer vision, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 38, 39

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus (2014). Intriguing properties of neural networks, *International Conference on Learning Representations (ICLR)*. 15, 108

Y. Taigman, M. Yang, M. Ranzato, and L. Wolf (2014). Deepface: Closing the gap to human-level performance in face verification, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 11, 12, 32, 34, 38, 39, 66, 86

M. Tan (1993). Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents, *Proceedings of the International Conference on Machine Learning (ICML)*. 173

E. Teh, M. Rochan, and Y. Wang (2016). Attention Networks for Weakly Supervised Object Localization, *Proceedings of the British Machine Vision Conference (BMVC)*. 152

F. Tramer, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart (2016). Stealing Machine Learning Models via Prediction APIs, *USENIX Security Symposium*. 19

L. van der Maaten and G. Hinton (2008). Visualizing High-Dimensional Data Using t-SNE, *Journal of Machine Learning Research*. 140

V. N. Vapnik and A. Y. Chervonenkis (2015). On the uniform convergence of relative frequencies of events to their probabilities, *Measures of complexity*, pp. 11–30. 9, 174

R. R. Varior, B. Shuai, J. Lu, D. Xu, and G. Wang (2016). A Siamese Long Short-Term Memory Architecture for Human Re-identification, *Proceedings of the European Conference on Computer Vision (ECCV)*. 11

A. Vezhnevets, V. Ferrari, and J. Buhmann (2011). Weakly Supervised Semantic Segmentation with a Multi-image Model, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 152

T. Vu, A. Osokin, and I. Laptev (2015). Context-aware CNNs for person head detection, *International Conference on Computer Vision (ICCV)*. 56

J. Walker, K. Marino, A. Gupta, and M. Hebert (2017). The Pose Knows: Video Forecasting by Generating Pose Futures, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 70

Y. Wang and K. N. Plataniotis (2010). An Analysis of Random Projection for Changeable and Privacy-Preserving Biometric Verification, *IEEE Transactions on Systems, Man, and, Cybernetics: part B: CYBERNETICS*. 86

Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli (2004). Image quality assessment: from error visibility to structural similarity, *IEEE Transactions on Image Processing (TIP)*, vol. 13(4), pp. 600–612. 77

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*. 150, 151, 152, 167

M. J. Wilber, V. Shmatikov, and S. Belongie (2016). Can We Still Avoid Automatic Face Detection?, *IEEE Winter Conference on Applications of Computer Vision (WACV)*. 13, 53, 57, 108

G. L. Wittel and S. F. Wu (2004). On Attacking Statistical Spam Filters, *Proceedings of the Conference on Email and Anti-Spam (CEAS)*. 15

W. X and G. A (2016). Generative Image Modeling Using Style and Structure Adversarial Networks, *Proceedings of the European Conference on Computer Vision (ECCV)*. 70

C. Xiao, J.-Y. Zhu, B. Li, W. He, M. Liu, and D. Song (2018). Spatially Transformed Adversarial Examples, *International Conference on Learning Representations (ICLR)*. 15, 173

T. Xiao, H. Li, W. Ouyang, and X. Wang (2016). Learning Deep Feature Representations With Domain Guided Dropout for Person Re-Identification, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 11

J. Xu, A. Schwing, and R. Urtasun (2015). Learning To Segment under Various Forms of Weak Supervision, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 149, 151, 152

W. Xu, D. Evans, and Y. Qi (2017). Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks, *arXiv*. 16

K. Yamada, Y. Sugano, T. Okabe, Y. Sato, A. Sugimoto, and K. Hiraki (2010). Can saliency map models predict human egocentric visual attention?, *Proceedings of the Asian Conference on Computer Vision (ACCV)*. 152, 158

A. C.-C. Yao (1986). How to Generate and Exchange Secrets, *IEEE Annual Symposium on Foundations of Computer Science (FOCS).* 90, 94

R. Yeh, C. Chen, T. Lim, M. Hasegawa-Johnson, and M. N. Do (2016). Semantic Image Inpainting with Perceptual and Contextual Losses, *arXiv.* 70

D. Yi, Z. Lei, and S. Z. Li (2014a). Deep Metric Learning for Practical Person Re-Identification, *International Conference on Pattern Recognition (ICPR).* 11

D. Yi, Z. Lei, S. Liao, and S. Z. Li (2014b). Learning Face Representation from Scratch, *arXiv.* 32, 38

F. Yu and V. Koltun (2016). Multi-Scale Context Aggregation by Dilated Convolutions, *International Conference on Learning Representations (ICLR).* 153

A. Zadeh, T. Baltrusaitis, and L. Morency (2017). Convolutional Experts Constrained Local Model for Facial Landmark Detection, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).* 72

M. D. Zeiler and R. Fergus (2014). Visualizing and Understanding Convolutional Networks, *Proceedings of the European Conference on Computer Vision (ECCV).* 151

X. Zeng, C. Liu, W. Qiu, L. Xie, Y.-W. Tai, C. K. Tang, and A. L. Yuille (2017). Adversarial Attacks Beyond the Image Space, *arXiv.* 15, 173

H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. N. Metaxas (2017). StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 70

J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff (2016). Top-Down Neural Attention by Excitation Backprop, *Proceedings of the European Conference on Computer Vision (ECCV).* 151, 154, 157

J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Měch (2015a). Minimum Barrier Salient Object Detection at 80 FPS, *Proceedings of the IEEE International Conference on Computer Vision (ICCV).* 159

N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. D. Bourdev (2014). PANDA: Pose Aligned Networks for Deep Attribute Modeling, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 86

N. Zhang, M. Paluri, Y. Taigman, R. Fergus, and L. Bourdev (2015b). Beyond Frontal Faces: Improving Person Recognition Using Multiple Cues, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 11, 12, 22, 23, 24, 26, 32, 34, 38, 39, 40, 48, 54, 55, 60, 66, 74, 116

S. Zhang, R. Benenson, and B. Schiele (2015c). Filtered channel features for pedestrian detection, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 24

R. Zhao, W. Ouyang, H. Li, and X. Wang (2015). Saliency detection by multi-context deep learning, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 153, 164

Z. Zhao, D. Dua, and S. Singh (2018). Generating Natural Adversarial Examples, *International Conference on Learning Representations (ICLR).* 15

E. Zheleva and L. Getoor (2009). To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles, *Proceedings of the ACM international conference on World wide web (WWW).* 13

B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba (2016). Learning Deep Features for Discriminative Localization., *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 151, 154, 164

B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva (2014). Learning Deep Features for Scene Recognition using Places Database., *Advances in Neural Information Processing Systems (NIPS).* 30, 31

X. Zhu and D. Ramanan (2012). Face detection, pose estimation and landmark localization in the wild, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 93