

©Copyright 2021

Ivan Evtimov

Disrupting Machine Learning:  
Emerging Threats and Applications for Privacy and Dataset  
Ownership

Ivan Evtimov

A dissertation  
submitted in partial fulfillment of the  
requirements for the degree of

Doctor of Philosophy

University of Washington

2021

Reading Committee:

Tadayoshi Kohno, Chair

Emre Kiciman

David Kohlbrenner

Franziska Roesner

Program Authorized to Offer Degree:  
Paul G. Allen School of Computer Science & Engineering

University of Washington

**Abstract**

Disrupting Machine Learning:  
Emerging Threats and Applications for Privacy and Dataset Ownership

Ivan Evtimov

Chair of the Supervisory Committee:  
Professor Tadayoshi Kohno  
Paul G. Allen School of Computer Science & Engineering

Convolutional neural networks (CNNs) can be trained with machine learning techniques by using large datasets of images to solve a multitude of useful computer vision tasks. However, CNNs also suffer from a set of vulnerabilities that allow maliciously crafted inputs to affect both their inference and training. A central premise of this dissertation is that these vulnerabilities exhibit a duality when it comes to security and privacy. On the one hand, when computer vision models are applied in safety-critical settings such as autonomous driving, it is important to identify failures that can be exploited by malicious parties early on so that system designers can plan for novel threat models. On the other hand, when machine learning models themselves are being used in a malicious or unauthorized manner, such vulnerabilities can be leveraged to protect data creators from harmful effects of these models (such as privacy degradation) and enforce finer-grained “access” controls over the data. This work studies security and privacy issues in three scenarios where machine learning is applied for visual tasks. The first contribution of this work is to identify a vulnerability in models that are likely to be deployed to identify road signs in autonomous vehicles. It demonstrates that an attacker with no digital access to a self-driving car’s computers can nevertheless cause dangerous behavior by modifying the appearance of physical objects. Next, this dissertation considers scenarios where machine learning models are applied in a way that degrades individual privacy. The dissertation proposes a scheme – nicknamed FoggySight – in which a community

of users volunteer adversarial modified photos (“decoys”) that poison the facial search database and throw off searches in it. Finally, machine learning models may be trained on data without authorization to do so. This dissertation considers scenarios where image owners might wish to share their visual data widely for human consumption but do not wish to enable its use for machine learning purposes. It develops a protective mechanism that can be applied to datasets before they are released so that unauthorized parties cannot train their models on them.

# TABLE OF CONTENTS

	Page
List of Figures . . . . .	iii
List of Tables . . . . .	vi
Chapter 1: Introduction . . . . .	1
1.1 Novel Vulnerabilities in Computer Vision Models Applied in Critical Settings . . . . .	2
1.2 Restoring Privacy from Facial Lookups . . . . .	2
1.3 Protecting Data From Unauthorized Use For Machine Learning . . . . .	3
1.4 Summary of Contributions . . . . .	4
Chapter 2: Background and Related Work . . . . .	7
2.1 Adversarial Examples . . . . .	7
2.2 Face Recognition . . . . .	9
2.3 Adversarial Examples and Face Recognition . . . . .	10
2.4 Other Attacks on Face Recognition Models . . . . .	13
2.5 Data Poisoning . . . . .	14
2.6 Biases and Other Causes of Failure in Training Models . . . . .	16
2.7 Miscellaneous and Orthogonal Work on Dataset Security and Privacy . . . . .	18
Chapter 3: Robust Physical-World Attacks on Deep Learning Visual Classification . . . . .	19
3.1 Introduction . . . . .	19
3.2 Adversarial Examples for Physical Objects . . . . .	21
3.3 Experiments . . . . .	25
3.4 Discussion . . . . .	30
Chapter 4: FoggySight: A Scheme for Facial Lookup Privacy . . . . .	32
4.1 Introduction . . . . .	32

4.2	Definition of Terms . . . . .	33
4.3	Goals and Assumptions . . . . .	34
4.4	The FoggySight Design . . . . .	36
4.5	Experimental Setup and Metrics . . . . .	44
4.6	Evaluation . . . . .	48
4.7	Discussion . . . . .	54
Chapter 5:	Disrupting Model Training with Adversarial Shortcuts . . . . .	58
5.1	Introduction . . . . .	58
5.2	Setup and Goals . . . . .	59
5.3	Proposed Methods for Protective Dataset Modifications . . . . .	60
5.4	Experimental Setup . . . . .	64
5.5	Evaluation . . . . .	64
5.6	Discussion . . . . .	67
Chapter 6:	Conclusion . . . . .	69
	Bibliography . . . . .	72
Appendix A:	Additional Tables and Figures for Robust Physical-World Perturbations . . . . .	87
Appendix B:	Additional Material for FoggySight . . . . .	96
B.1	Adversarial Examples Success . . . . .	96
B.2	Alternative Targeting Mechanisms . . . . .	96
B.3	Solo Action Defenses with Untargeted Adversarial Examples . . . . .	100
Appendix C:	Additional Figures for Disrupting Unauthorized Uses of Machine Learning . . . . .	105

## LIST OF FIGURES

Figure Number	Page
<p>2.1 Simplified visual representation of the metric space learned by state-of-the-art face recognition neural networks. When a tightly cropped facial image is processed by a neural network, it produces an embedding vector (here, represented by a dot in <math>\mathbb{R}^2</math>). Pairs of vectors belonging to different identities are far away from each other while those belonging to the same identity are close together. In practice, neural networks produce vectors in <math>\mathbb{R}^{128}</math> or <math>\mathbb{R}^{512}</math> and metrics such as Euclidean distance and cosine similarity define “closeness.” . . . . .</p>	11
<p>3.1 The left image shows real graffiti on a Stop sign, something that most humans would not think is suspicious. The right image shows our a physical perturbation applied to a Stop sign. We design our perturbations to mimic graffiti, and thus “hide in the human psyche.” . . . . .</p>	20
<p>3.2 <math>RP_2</math> pipeline overview. The input is the target Stop sign. <math>RP_2</math> samples from a distribution that models physical dynamics (in this case, varying distances and angles), and uses a mask to project computed perturbations to a shape that resembles graffiti. The adversary prints out the resulting perturbations and sticks them to the target Stop sign. . . . .</p>	21
<p>4.1 Visual illustration of the FoggySight privacy defense strategy. Decoy photos are pictures belonging to different identities that are adversarially modified so that face recognition neural networks produce embedding vectors close to those of the identity being protected (denoted as “A”). Therefore, decoy photos appear as the closest neighbors of a query photo of A and the real identity is not revealed in response to the query. . . . .</p>	39
<p>4.2 Plots of privacy strategy success when targeting a randomly sampled lookup set vector. Observe that perturbation magnitudes of <math>\epsilon \geq 0.06</math> achieve low recall, low discovery, and high identity uniformity, thereby successfully preserving the privacy of the protected individuals. . . . .</p>	49
<p>4.3 Plots of privacy strategy success when targeting the mean of the lookup set. While this defense leaks less information to the protectors and is easier to coordinate, it does not achieve results as good as when targeting a randomly sampled lookup set photo. . . . .</p>	50

4.4	Graphs of privacy strategy success versus the number of decoy set photos. . . . .	52
4.5	Experimental results when protectors do not have access to the face recognition model	54
4.6	Illustration of final transferable decoy images under different perturbation magnitudes $\epsilon$ . These are images of subject n000957 in the VGGFace2 dataset modified to serve as decoys for other identities. . . . .	55
5.1	Example of an ImageNet-sized image with the various modification techniques applied. The image depicted here was originally available at <a href="https://www.flickr.com/photos/volvob12b/9797687423">https://www.flickr.com/photos/volvob12b/9797687423</a> , was accessed on June 3, 2021, and is distributed in the Public Domain. To the best of our knowledge, this image is not actually part of the ImageNet dataset but if it were, it would have class index 263 for ‘Pembroke, Pembroke Welsh corgi.’ . . . . .	62
5.2	Best achievable test accuracy after 50 epochs when training ResNet18 on CIFAR-10 with different dataset modifications. . . . .	65
5.3	Validation accuracy progress when training ResNet18 on a protected ImageNet with standard augmentations during training. . . . .	68
B.1	Magnitude of final optimization loss after decoy photo generation under different perturbation magnitudes $\epsilon$ . Note that the case where $\epsilon = 0.0$ corresponds to the unmodified photos. As expected, the higher the perturbation amount, the better the PGD algorithm for adversarial examples generation achieves its goal. . . . .	97
B.2	Illustration of final decoy images under different perturbation magnitudes $\epsilon$ . These are images of subject n000029 in the VGGFace2 dataset modified according to the “randomly sampled target from the lookup set” strategy to produce vectors in the region of subject n000958. . . . .	98
B.3	Privacy strategy success when targeting the same photo of the protected user universally. All results averaged over all identities and all photos. While this strategy does manage to bring recall down, it is less effective at reducing the discovery rate and the uniformity of identities in the top recall set. . . . .	99
B.4	Graphs of privacy strategy success when targeting a sample from a Gaussian model. Observe that this scheme fares just as well as when targeting the mean lookup set by comparing with Figure 4.3. . . . .	100
B.5	A visual illustration of the solo action defense. A user aims to shift his or her face images far away from their original location in the embedding space. This fills the recall set with other identities. . . . .	102



B.6	Recall and discovery rate at various levels of $k$ and $\epsilon$ when assuming the protected has 100% control of their own lookup set. The perturbation amount is normalized to represent percentage relative to standard deviation (images have unit standard deviation). For both metrics, a perturbation amount of 0.04 suffices to evade recognition. “Top Hits” refers to the recall set of nearest neighbors to the query photo that is returned by the facial search service to its user. . . . .	103
B.7	Recall and discovery rate at various levels of $k$ and $\epsilon$ when assuming the protected only has limited control of their own lookup set (as controlled by the subsample rate). The perturbation amount is normalized to represent percentage relative to standard deviation (images are have unit standard deviation). Only having access to a fraction of the lookup data drastically degrades privacy protection. This indicates that other strategies are needed in the case that we cannot modify 100% of the target’s data. “Top Hits” refers to the recall set of nearest neighbors to the query photo that is returned by the facial search service to its user. . . . .	104
C.1	Results on training a modified CIFAR10 with no countermeasures. . . . .	106
C.2	Results when applying countermeasures to the modified CIFAR10 training set. . .	107
C.3	Examples of the perturbed CIFAR10 training set with a pixel-based perturbation approach at various settings of the parameter $\mu$ . . . . .	108
C.4	Examples of the perturbed CIFAR10 training set with a visual watermarking approach at various settings of the parameter $\alpha$ . . . . .	109
C.5	Examples of the perturbed CIFAR10 training set with a brightness modulation approach at various settings of the parameter $\gamma$ . . . . .	110

## LIST OF TABLES

Table Number	Page
3.1 Sample of physical adversarial examples against LISA-CNN and GTSRB-CNN. . .	31
A.1 Targeted physical perturbation experiment results on LISA-CNN using a poster-printed Stop sign (subtle attacks) and a real Stop sign (camouflage graffiti attacks, camouflage art attacks). For each image, the top two labels and their associated confidence values are shown. The misclassification target was Speed Limit 45. See Table 3.1 for example images of each attack. Legend: SL45 = Speed Limit 45, STP = Stop, YLD = Yield, ADL = Added Lane, SA = Signal Ahead, LE = Lane Ends. . .	88
A.2 Poster-printed perturbation (faded arrow) attack against the LISA-CNN for a Right Turn sign at varying distances and angles. See example images in Table 1 of the main text. Our targeted-attack success rate is 73.33%. . . . .	89
A.3 Drive-by testing summary for LISA-CNN. In our baseline test, all frames were correctly classified as a Stop sign. We have added the yellow boxes as a visual guide manually. . . . .	90
A.4 A camouflage art attack on GTSRB-CNN. See example images in Table 3.1. The targeted-attack success rate is 80% (true class label: Stop, target: Speed Limit 80).	91
A.5 Sticker perturbation attack on the Inception-v3 classifier. The original classification is microwave and the attacker’s target is phone. See example images in Table A.7. Our targeted-attack success rate is 90% . . . . .	92
A.6 Sticker perturbation attack on the Inception-v3 classifier. The original classification is coffee mug and the attacker’s target is cash machine. See example images in Table A.8. Our targeted-attack success rate is 71.4%. . . . .	93
A.7 Uncropped images of the microwave with an adversarial sticker designed for Inception-v3. . . . .	94
A.8 Cropped Images of the coffee mug with an adversarial sticker designed for Inception-v3. . . . .	95

## ACKNOWLEDGMENTS

Pursuing a PhD – especially for someone who did not have early academic role models – was never a given for me; I owe the fact that I have completed one to a multitude of people.

First and foremost, my advisor Tadayoshi (Yoshi) Kohno played a tremendously important role. Yoshi provided me with the freedom to explore and grow and coupled that with an unwavering support and encouragement for my academic endeavors. Thinking back to all my successes in grad school, they all began with an introduction, an idea, or an advice from Yoshi. Thinking back to all of my failures, stumbles and moments of doubt, they were all met with a seemingly endless well of support and encouragement by Yoshi. Undoubtedly, Yoshi's determination to believe in me even when I did not believe in myself was the major reason I could persevere, grow and accomplish everything I have accomplished. He was and is the best a mentor and an advisor can hope to be.

One of the first introductions Yoshi ever made was to Earlence Fernandes. Earlence brought me onto the project that would become our most famous and highly cited work, entrusted me with a leading role in the infancy of that project, and took it upon himself to set my academic career on a good path from early on. He treated me like an equal, introduced me to the lay of the obscure land of academia, and – most importantly – hauled stop signs with me when stop signs needed to be hauled. Thank you, Earlence, for being such a great mentor to me!

A mentor who worked closely with Yoshi and was equally as supportive and encouraging is Franzi Roesner. I only wish we had had a chance to collaborate more. Yoshi and Franzi are also responsible for building up the Security and Privacy Lab, which was another important source of support, ideas, and fun for me. Thank you to Christine Chen, Kaiming Cheng, Camille Cobb, Miro Enev, Christine Geeng, David Kohlbrenner, Karl Koscher, Kiron Lebeck, Shirang Mare, Peter Ney, Kentrell Owens, Kimberly Ruth, Lucy Simko, Anna Kornfeld Simpson, Alex Takakuwa, Miranda

Wei, and Eric Zeng. I will miss the endless discussions and idea-generating sessions we had (and ski day).

None of the works in this dissertation would have been possible without the contributions of my collaborators. On the material in Chapter 3, I had the fortune of collaborating with Kevin Eykholt, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Dawn Song, and Florian Tramèr, in addition to my advisors, who I already mentioned. That this work has gone on to be known and used as widely as it is, is a testament to our incredibly successful team effort. In the development of FoggySight (Chapter 4), Pascal Sturmfels brought his expertise in machine learning and set up the early framework of the work. On the topic of Chapter 5, I received particularly helpful guidance from and had many fruitful conversations with Ian Covert and Aditya Kusupati. I also thank Gabriel Ilharco and Samuel Ainsworth in addition to my committee members for providing their feedback and thoughts on that study.

My professional growth was also aided by several internships I completed. At Microsoft Research, Ece Kamar, Emre Kiciman, Jerry Li, Weidong Cui, and Eric Horvitz, along with collaborators who I cannot name for the sake of secrecy, had set me up to succeed at a project that was incredibly exciting and important (and will remain equally unspoken of). At Facebook, I had the pleasure of collaborating with Russ Howes, Brian Dolhansky, Hamed Firooz, Cristian Canton Ferrer, Aaron Jaech and others. I look forward to many more successful projects on advancing the state of the art in adversarial machine learning research in my new role on the AI Red Team alongside some of those folks. I am also grateful to Ryan Calo and David O’Hair, who helped educate me and the broader world about how the law handles hacking stop signs. I would also like to thank Ryan for being another source of support and for multiple incredibly interesting discussions.

I also thank the members of the committee for their feedback and guidance and the one member who I have not named yet in another capacity is Mary Fan. Thank you, Mary, for being a great GSR and offering important advice on my generals and thesis!

There are also several people at the Allen School and at the University of Washington, who

made my life and research much easier. Elise Dorough is an absolute superhero, whose special powers keep everything in the Allen School moving incredibly smoothly and has saved me countless hours and anxiety. Melody Kadenko not only brought things forward at a lightning speed when we needed to purchase the occasional stop sign or GPU rig but also was happy to share stories from Ukraine and hear mine from Bulgaria. I regret she has moved on to other endeavors but I am glad Dali Grubisa was there to take over for her. Emily McReynolds and Hannah Almeter from the Tech Policy Lab also had an impact on me through the Tech Policy Lab.

My time in the PhD – and especially the tough moments of grad school and my personal life – was made much easier by an incredibly supportive group of friends. In addition to those I have mentioned already, I thank Ofir Press, Victor Zhong, Judit Acs, Gabor Szabo, Sally Dong, Nick Nuechterlein, Erin Wilson, Gabe Erion, and many others for all the fun we had and their amazing friendships with me.

My graduate school work would not have been possible without the foundation from previous educational institutions. I thank my English teacher, Ilian Iliev, who first encouraged me to dream big and think about going to college on the other end of the world. I also thank Professors Amir Sadovnik, Chris Ruebeck, Jonathan Lafky, and Chawne Kimber, who opened up the world of academia to me back at Lafayette College.

Last but not least, my parents, Mima and Petar Evtimovi, and my grandparents, Mariika and Ivan Evtimovi, provided me with loving support and freedom to make bold choices in life, beyond what anybody in the family had ever done.

### **Papers Included in This Dissertation.**

The material in Chapter 3, in Appendix A, and in portions of Chapters 1, 2, and 6 appears in the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. I was a co-first author of this work. The full citation for this work is:

Eykholt, Kevin, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. “Robust physical-world attacks on deep learning visual classification.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1625-1634. 2018.

The material in Chapter 4, in Appendix B, and in portions of Chapters 1, 2, and 6 is due to appear in Proceedings on Privacy Enhancing Technologies 2021 (3). I was a first author of this work. The full citation for this work is:

Evtimov, Ivan, Pascal Sturmfels, and Tadayoshi Kohno. “FoggySight: A Scheme for Facial Lookup Privacy.” In *Proceedings on Privacy Enhancing Technologies*, vol.2021, no.3, 2021, pp. 204-226.

The material in Chapter 5, in Appendix C, and in portions of Chapters 1, 2, and 6 appears as a pre-print on the arXiv distribution service. I was a first author of this work. The full citation for this work is:

Evtimov, Ivan, Ian Covert, Aditya Kusupati, and Tadayoshi Kohno. “Disrupting Model Training with Adversarial Shortcuts.” *arXiv preprint arXiv:2106.06654* (2021). <https://arxiv.org/abs/2106.06654>

### **Funding Acknowledgements.**

My work was supported in part by the University of Washington Tech Policy Lab, which receives support from: the William and Flora Hewlett Foundation, the John D. and Catherine T. MacArthur Foundation, Microsoft, the Pierre and Pamela Omidyar Fund at the Silicon Valley Community Foundation; it was also supported by the US National Science Foundation (Award 156525). In

addition, the work in Chapter 3 was supported in part by NSF grants 1422211, 1616575, 1646392, 1740897, 1565252, Berkeley Deep Drive, the Center for Long-Term Cybersecurity, and FORCES (which receives support from the NSF).

## **DEDICATION**

To my parents, Mima and Petar Evtimovi, and my grandparents, Mariika and Ivan Evtimovi and Liliana Rusanova, who nurtured my curiosity and perseverance from an early age and never wavered in supporting me, even as I ventured thousands of kilometers away from them.



## Chapter 1

### **INTRODUCTION**

Convolutional Neural Networks (CNNs) have been the cornerstone of a series of important advances in computer vision. In classification, they can correctly predict what type of object is depicted in a picture out of a thousand possibilities with accuracy above 80% [52]. In detection and segmentation, they can determine the location, shape, and class of objects with high precision [111, 112]. And in face recognition, they can identify the correct identity of an individual in a photo from a set of 12 million identities with error rates less than 0.2% [48], far surpassing human performance [72]. This success on standard computer vision tasks has led to their deployment in a broad array of areas of human activity. To name just a few, CNNs are used for perception and control in cyber-physical systems such as cars [40, 81], UAVs [14, 96], and robots [155], for analyzing medical images [91, 101], and for filtering hateful, harmful, or illegal content on social networks [70, 139, 144].

As with any broadly adopted technology, CNNs have created a new set of security and privacy challenges. Perhaps more so than with other technologies, however, these challenges exhibit an important duality. On the one hand, when CNNs are used in software with safety-critical functions, their proper and reliable operation is an asset to be protected. On the other hand, CNNs are often used with detrimental consequences to individual privacy or against the will of the owners of data used to train them. In those cases, these models become a threat themselves and vulnerabilities can be exploited to reduce harm stemming from their operation.

This dissertation contributes three studies that each resolve an important security and privacy issue related to the processing of visual data with CNNs.

### ***1.1 Novel Vulnerabilities in Computer Vision Models Applied in Critical Settings***

As with many computer program, CNNs and machine learning models in general are vulnerable to exploitation through maliciously crafted inputs that cause mistakes in their operation. A particular class of such inputs is known as “adversarial examples” [20, 46, 136] and has garnered a lot of attention in the research community. In addition to inducing errors in the operations of models, adversarial examples are often surreptitious and hard to distinguish from legitimate inputs.

When CNNs are applied in the perception pipeline of cyberphysical systems, this creates a potential threat from a new class of adversaries. Attackers who do not have digital access to a system nevertheless might be able to influence its behavior by modifying the physical world and placing adversarial examples that mislead the computer vision component of such a system. Consider the case of autonomous driving. In this situation, it is critically important for the computer of the car to identify what road signs are in front of it so that it can make correct driving decisions.

Chapter 3 tackles the question of whether it is possible to induce errors in the classification of road signs solely by modifying the physical object in a hard-to-detect manner. It answers that question by developing a new algorithm – Robust Physical Perturbations (RP<sub>2</sub>) – that can produce physical alterations to objects that consistently induce mistakes in road sign recognition models and in state-of-the-art object classification models. As one particularly evocative example, RP<sub>2</sub> can produce stickers that, when applied to stop signs, cause models to classify them as speed limit signs. Robust experiments in different viewing conditions show that this threat is to be taken seriously and that relying on the variations from the real world is not enough to prevent exploitation.

### ***1.2 Restoring Privacy from Facial Lookups***

While Chapter 3 focuses on security issues when it is desirable that computer vision models operate well in the presence of adversaries, this is not always the case. The problem of ubiquitous facial search has recently become acute with the appearance of services such as ClearView [55, 57] and PimEyes [51] that link photos of individuals to their social media identities and other online presences. This is achieved by building up a database of facial photos associated with profiles on

websites such as Facebook, Twitter, and even Venmo. A photo taken from anyone anywhere can then be processed by a CNN face recognition model to match it up with the photos in the database. When such services are used by law enforcement, this raises a host of civil liberties questions around involuntary inclusion in criminal databases and reasonable search [33]. When they are available to the broader public, more nefarious applications of this easily accessible technology become possible, such as stalking and doxxing.

Therefore, the “proper” operation of CNN models in that case is undesirable for many individuals whose privacy may suffer from facial searches. Chapter 4 proposes a methodology by which the vulnerabilities in machine learning models can be leveraged effectively in order to regain privacy against ubiquitous facial searches. The approach is named FoggySight and is meant to poison the database used for facial search in a collaborative manner such that existing photos of individuals are crowded out by “decoys” (adversarial examples). This work studies the conditions needed for FoggySight to be successful and finds that individuals can meaningfully increase their privacy when other “protectors” feed adversarially modified photos (“decoys”) in the facial database.

### ***1.3 Protecting Data From Unauthorized Use For Machine Learning***

Finally, there is a third scenario with unique security problems stemming from the application of machine learning that require a finer-grained access control mechanism for visual data. There are two different levels of privileged access to large collections of images: (1) semantic understanding access – the ability of a human to view and understand images in a dataset; and (2) machine learning access – the ability to train machine learning models on that data. A lot of research and several mature fields of computer science have developed mechanisms that allow for machine learning access while withholding semantic understanding access. For example, differential privacy and homomorphic encryption may allow a model to be learned by an untrusted party that is not given access to the raw image data. However, no mechanism exists to grant semantic understanding access while withholding machine learning access.

Chapter 5 work proposes and thoroughly evaluates a method for disrupting machine learning training while maintaining the semantic quality of images. It proposes three different approaches,

studies their feasibility in preventing training on common computer vision benchmarks, and compares the effectiveness of each against previously published, gradient-based methods. Among other conclusions, the findings increase our understanding of how CNN training fails in the presence of simple correlations in the training set. A central finding of this work is that CNNs prefer to fit a broad variety of simple patterns to the true semantics of the task at hand.

#### **1.4 Summary of Contributions**

Here, we summarize the findings of each of the chapters.

In Chapter 3:

- We introduce Robust Physical Perturbations ( $RP_2$ ) to generate physical perturbations for *physical-world* objects that can consistently cause misclassification in a CNN-based classifier under a range of dynamic physical conditions, including different viewpoint angles and distances (Section 3.2).
- Given the lack of a standardized methodology in evaluating physical adversarial perturbations, we propose an evaluation methodology to study the effectiveness of physical perturbations in real world scenarios.
- We evaluate our attacks against two standard-architecture classifiers that we built: LISA-CNN with 91% accuracy on the LISA test set and GTSRB-CNN with 95.7% accuracy on the GTSRB test set. Using two types of attacks (object-constrained poster and sticker attacks) that we introduce, we show that  $RP_2$  produces robust perturbations for real road signs. For example, poster attacks are successful in 100% of stationary and drive-by tests against LISA-CNN, and sticker attacks are successful in 80% of stationary testing conditions and in 87.5% of the extracted video frames against GTSRB-CNN.
- To show the generality of our approach, we generate the robust physical adversarial example by manipulating general physical objects, such as a microwave. We show that the pre-trained Inception-v3 classifier misclassifies the microwave as “phone” by adding a single sticker.

In Chapter 4:

- We propose FoggySight: a collaborative facial privacy approach meant to poison the database used for facial search. We study the conditions needed for FoggySight to be successful and find that individuals can meaningfully increase their privacy when other “protectors” feed adversarially modified photos (“decoys”) in the facial database.
- We compare and evaluate different approaches for generating adversarial examples/decoys in the metric learning space defined by face recognition neural networks and find the most effective approach to be to target the mean of an individual’s available facial vectors. In that scenario, protected individuals only need protectors to provide decoys numbering 2-4 times the number of unmodified photos of the protected, when protectors have access to the facial search model.
- We study the effectiveness of FoggySight when protectors do not have access to the facial search model. In those cases, protectors need to increase both the magnitude of modifications in the decoys and the number they provide relative to the clean photos of the protected. But they can still meaningfully increase the privacy of the protected: under the right parameters, we show that our scheme can decrease the identification rate on the Azure Face Service to under 10%.

In Chapter 5:

- We introduce the notion of adversarial shortcuts and propose three dataset modification techniques to prevent CNNs from learning useful classification functions.
- We evaluate each technique on the popular CIFAR-10 [76] and ImageNet [113] datasets and find that the proposed techniques severely limit the test-set accuracy of state-of-the-art models. We also verify that our techniques are robust to certain simple countermeasures.
- We compare our approach to a concurrent proposal [37] and show that our simpler approach based on adversarial shortcuts proves more effective at disrupting model training.

Each of these studies increase our understanding of security and privacy issues in machine learning-based computer vision. These contributions also provide the basis for further research into securing models when they applied benignly and disrupting their operation or creation when they are not.

## Chapter 2

### BACKGROUND AND RELATED WORK

This chapter provides definitions of concepts related to the work in the following chapters. We begin with an overview of the work on adversarial examples that is the basis for the study in Chapter 3. Next, we introduce terms and concepts specific to face recognition as it relates to Chapter 4 and summarize work on disrupting face recognition with adversarial machine learning. Finally, we summarize relevant work on how modifications to datasets impact security and privacy during model training and describe how those studies relate to Chapter 5.

#### 2.1 Adversarial Examples

While machine learning models in general have long been known to be vulnerable to adversarial inputs [25, 85, 86], the majority of recent work has focused on the study of *adversarial examples*. Given a classifier  $f_\theta(\cdot)$  with parameters  $\theta$  and an input  $x$  with ground truth label  $y$  for  $x$ , an adversarial example  $x'$  is generated so that it is close to  $x$  in terms of certain distance, such as  $L_p$  norm distance.  $x'$  will also cause the classifier to make an incorrect prediction as  $f_\theta(x') \neq y$  (untargeted attacks), or  $f_\theta(x') = y^*$  (targeted attacks) for a specific  $y^* \neq y$ .

##### 2.1.1 Digital Adversarial Examples.

Adversarial examples were first observed in [136]. More sophisticated approaches to develop them followed quickly [46, 95] and the literature since then has followed an attack/defense cycle. While multiple works have put forward techniques to disrupt the adversarial nature of such inputs, nearly all of them have been followed by “attack” papers that produce stronger adversarial examples. This led to several “standard” methods to develop adversarial examples: optimizing a regularized loss function with standard gradient descent [20] and using projected gradient descent [88]. Adaptations

of those approaches have repeatedly defeated state-of-the-art “defenses” [8, 19, 20, 140].

Much research work has also observed that adversarial examples can be generated without knowledge of the internals of the model simply by querying it [64, 106] or by attacking a similar model and relying on transferability, the ability of adversarial examples that are effective against one model to mislead another one [83].

In short, adversarial examples are fundamental vulnerabilities in DL models that have not been remedied reliably to this day. They allow their creators to control the output of neural network models while preserving visual similarity to non-adversarial images.

In Chapter 3, we focus on the setting where adversaries have full access for two reasons: (1) In our chosen autonomous vehicle domain, an attacker can obtain a close approximation of the model by reverse engineering the vehicle’s systems using model extraction attacks [141]. (2) To develop a foundation for future defenses, we must assess the abilities of powerful adversaries, and this can be done in a white-box setting. Given that recent work has examined the transferability of digital adversarial examples [105], physical query-only or transferable attacks may also be possible. In Chapter 4, we examine transferability for adversarial examples against face recognition models.

### *2.1.2 Physical Adversarial Examples.*

Kurakin et al. showed that printed adversarial examples can be misclassified when viewed through a smartphone camera [78]. Athalye and Sutskever improved upon the work of Kurakin et al. and presented an attack algorithm that produces adversarial examples robust to a set of two-dimensional synthetic transformations [7]. These works do not modify physical objects—an adversary prints out a digitally-perturbed image on paper. However, there is value in studying the effectiveness of such attacks when subject to environmental variability. Our object-constrained poster printing attack is a reproduced version of this type of attack, with the additional physical-world constraint of confining perturbations to the surface area of the sign. Additionally, our work goes further and examines how to effectively create adversarial examples where the object itself is physically perturbed by placing stickers on it.



Concurrent to our work,<sup>1</sup> Athalye et al. improved upon their original attack, and created 3D-printed replicas of perturbed objects [9]. The main intellectual differences include: (1) Athalye et al. *only* use a set of synthetic transformations during optimization, which can miss subtle physical effects, while our work samples from a distribution modeling both physical *and* synthetic transformations. (2) Our work modifies *existing* true-sized objects. Athalye et al. 3D-print small-scale replicas. (3) Our work simulates realistic testing conditions appropriate to the use-case at hand.

Sharif et al. attacked face recognition systems by printing adversarial perturbations on the frames of eyeglasses [123]. Their work demonstrated successful physical attacks in relatively stable physical conditions with little variation in pose, distance/angle from the camera, and lighting. This contributes an interesting understanding of physical examples in stable environments. However, environmental conditions can vary widely in general and can contribute to reducing the effectiveness of perturbations. Therefore, we choose the inherently unconstrained environment of road-sign classification. In our work, we explicitly design our perturbations to be effective in the presence of diverse physical-world conditions (specifically, large distances/angles and resolution changes).

Finally, Lu et al. performed experiments with physical adversarial examples of road sign images against *detectors* and show current detectors cannot be attacked [87]. In this work, we focus on *classifiers* to demonstrate the physical attack effectiveness and to highlight their security vulnerability in the real world. Attacking detectors are out of the scope of this paper, though recent work has generated digital adversarial examples against detection/segmentation algorithms [24, 92, 151], and our recent work has extended  $RP_2$  to attack the YOLO detector [129].

## 2.2 Face Recognition

Automated face recognition has had a long history in the computer vision community [15]. Some of the earliest approaches to face recognition made use of basis decompositions [54, 142], local binary patterns [4] and SIFT features [11]. More recent approaches have made use of deep neural networks

---

<sup>1</sup>This work appeared at arXiv on 30 Oct 2017.

to automatically classify faces into known identities [132, 138]. These approaches are limited to only being able to classify faces from a known, preset list (e.g., the faces the model was trained on). To overcome this, state of the art approaches have cast face recognition as a metric learning problem. In this view, the goal is to learn an embedding space in which two faces of the same person are close and two faces of different people are far away. There exist many proposed loss functions to learn such an embedding space, including paired [59, 133] and triplet losses [56, 107, 117] — which directly optimize distance between pairs of faces — and clustering or max-margin style losses [30, 82, 146, 148], which aim to classify faces with an additive or multiplicative margin.

This more modern paradigm of metric learning differs from traditional classification in that the neural network models don't produce direct identity predictions. Rather, they produce embedding vectors of each input image such that images belonging to a given identity are clustered in the embedding space (see Fig. 2.1). This allows rapid face verification and lookup for identities that are not necessarily included in the network's training set via  $k$ -nearest neighbors.

A modern pipeline for face recognition using such a neural network might look as follows. First, the face recognition company either downloads a pre-trained, publicly available neural network designed for face recognition, or trains one themselves on an existing dataset where the identities are labeled. Then, they scrape the internet for publicly available photos from social media. They release an application combining their dataset and network. A user of the app takes a photo of a stranger in public. That photo is uploaded to the face recognition company's server, where it is cross-referenced against the photos collected from social media websites. The most similar faces according to the neural network are returned to the user of the app, along with the associated social media profiles. This is the approach used by the companies described in [55].

### **2.3 Adversarial Examples and Face Recognition**

There exist many works demonstrating the vulnerability of deep learning face recognition systems to *adversarial examples*. One set of works seeking to fool facial recognition models has focused on creating physical adversarial examples in the form of objects – such as glasses frames [123, 124] and hats [74] – that change the output of a model processing images of a person wearing them. Others

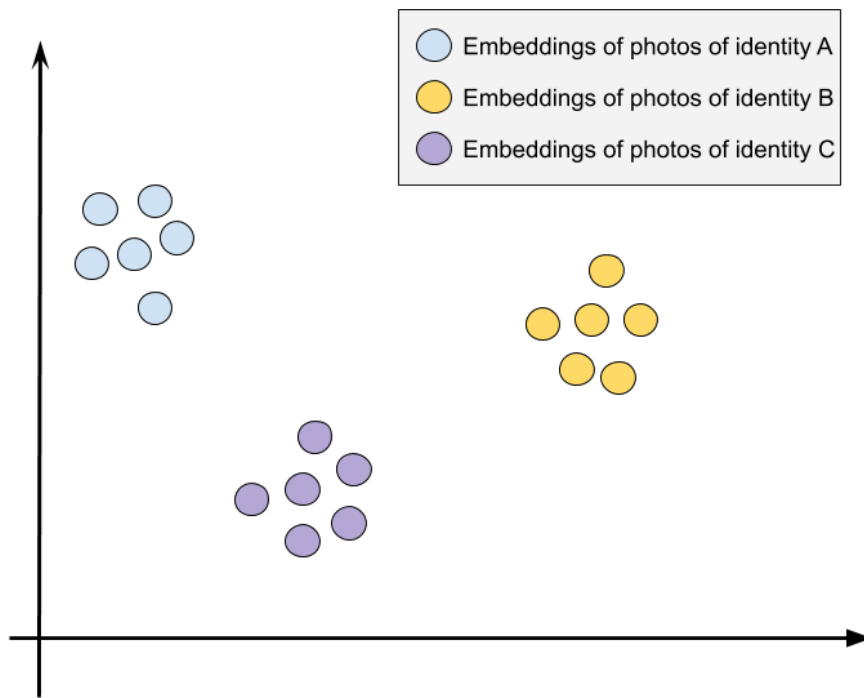


Figure 2.1: Simplified visual representation of the metric space learned by state-of-the-art face recognition neural networks. When a tightly cropped facial image is processed by a neural network, it produces an embedding vector (here, represented by a dot in  $\mathbb{R}^2$ ). Pairs of vectors belonging to different identities are far away from each other while those belonging to the same identity are close together. In practice, neural networks produce vectors in  $\mathbb{R}^{128}$  or  $\mathbb{R}^{512}$  and metrics such as Euclidean distance and cosine similarity define “closeness.”

showed that generating adversarial examples is also possible without possession of the weights and biases of the neural network but only with query-based oracle-like access to the model [31].

Of particular interest are works by [39] and [23] that develop transferable adversarial examples by optimizing in the metric space of facial recognition networks and study how much distortion individuals are willing to accept in their photos. In addition, [110] and [102] develop new approaches for generating adversarial examples against facial recognition that do not rely on the “standard” methods from [20, 88] and show that they are robust even in the face of countermeasures.

These works certainly indicate that adversarial examples and adversarial objects are particularly attractive mechanisms for protecting privacy from facial recognition. However, if individuals are to act as “attackers” of the neural network under the assumptions of the literature so far, they should be able to modify the query photo used to perform the search, or change their own appearance permanently. Neither of these is possible in a real-world scenario. Individuals can hardly control the photos others take of them. Anybody can snap a picture of anybody in a public space and CCTV and well-meaning web cams are pervasive. Furthermore, wearing adversarial accessories – such as hats and glasses – is not always practical or fashionable and restricts the individual’s freedom to control their own appearance. This is why we explore a scheme that does not assume control of the photo used to de-anonymize the individual.

A concurrent conference submission by [121] explores a similar solution to ours. This proposal, named Fawkes, also uses adversarial examples – named “cloaks” in that work – to disrupt the performance of facial classification. Cloaks have the same purpose as decoys in our work and are like adversarial examples from the adversarial machine learning literature. The authors also discuss a “Sybil attack” which corresponds to our communal defense strategy in which protector users upload cloaks/decoys/adversarial examples modified so that facial recognition models output a vector or classification corresponding to another user. Our work adds additional perspective by exploring what vector targets are best to use by the protectors and by applying an alternative transferable adversarial examples generation mechanism. In Section 4.4.3, we propose a number of possible mechanisms to select vectors in the metric learning space to use as targets for protectors and discuss their tradeoffs; in Section 4.6, we evaluate and compare those different approaches quantitatively. Furthermore, transferable cloak generation in Fawkes requires the protectors to use a robust neural network model trained on adversarial examples. In Section 4.4.2, we discuss an alternative method that does not require retraining and uses “out-of-the-box” models available online; we evaluate this method and find it to be successful in Section 4.6.3. A tradeoff of our method relative to Fawkes is that it requires larger perturbations to achieve privacy protections. Together, [121] and our paper provide a robust foundation for protecting face recognition under our shared threat model.

More broadly, our line of inquiry also fits in with studies on applying adversarial machine

learning for beneficial goals, such as [5, 28, 58, 145]. In most adversarial machine learning research, the party performing adversarial modifications is often referred to as the adversary. However, looking toward Section 4.3, we note that in our work – as in these other works – the party performing these modifications is not the adversary, but the party seeking defense against an adversary. Because FoggySight’s participants perform these “attacks” against the adversary’s capabilities, we may sometimes use the word “attack” to refer to the actions of the privacy protectors.

#### **2.4 Other Attacks on Face Recognition Models**

There exists a limited amount of work that attempts to fool face recognition systems by modifying photos at *training time* rather than at test time. [22] introduce a set of data poisoning attacks that modify a small number of the training photos in a face dataset. They show that a model trained on the poisoned dataset learns a back-door key: a pattern that, when presented to the model, gets the model to categorize that pattern as belonging to a particular face for impersonation purposes. They further demonstrate that they can instantiate this back-door key in the physical world by making the learned pattern a specific pair of glasses. Not specific to face recognition, there exists a body of work on attacking neural network systems using data poisoning attacks [49, 84, 97, 120]. A broader survey of data poisoning and backdooring attacks is given in [45].

There exists a subset of work on designing face recognition systems to be private [34, 89, 115, 150]. Those, in turn, are similar to work aiming to preserve the privacy of training set members and individual features of training set examples in machine learning more broadly [68, 69]. These works aim to design machine learning systems that don’t expose the model or database or (features of the) training set to the user and don’t expose the user to the model or server running the model. We view these works as tangential to ours: they still aim to design systems that are fundamentally able to identify individuals. Our main goal is to thwart such systems, with the assumption that those employing face recognition technology are not interested in our privacy.

Finally, the computer vision community has developed multiple approaches to anonymization that do not preserve the content of the original photo for human viewers – including some that apply adversarial modifications [80]. Those approaches are best used when stronger privacy guarantees

are required, such as when humans – and not just facial search services – are not supposed to be able to deidentify the individual in the photo. Therefore, we believe they are orthogonal to this work, as we aim to allow individuals to continue to derive utility from their facial photos.

## **2.5 Data Poisoning**

In Chapter 5, we are interested in “attacks” that happen at training time; we seek to modify the training set of a model in order to degrade its performance at inference time to an unusable level. The closest directions of research to that goal are those of data poisoning and backdooring. A useful recent survey of these attacks is provided in [45]; additionally, [118] aim to standardize the measurement and evaluation of successful attacks of this nature. In backdooring attacks, the adversary aims to tamper with the training procedure to produce a model that performs well on most of its test data at inference but when presented with specific instances with a specific trigger, the model’s behavior changes in a way chosen by the attacker. For example, [49] develops a method to influence the training of road sign classification models such that when they are presented with a stop sign with a sticky note, those models produce a wrong prediction (not a stop sign). This is achieved by modifying the training set to include examples that shift the trained model’s behavior on inputs with the trigger. These methods have been refined and extended in subsequent work, such as [22, 116, 143]. In [84], the authors study an attacker that can train the model themselves and provides it for reuse publicly (a common practice in machine learning research) and call this threat model “trojaning.” Backdooring and trojaning attacks are distinct from the methods we seek to develop in Chapter 5 in that we do not assume that the adversary controls any inputs to the model at inference time. Instead, we seek to degrade model performance at inference time on any unmodified input.

In Chapter 5, our goal and level of adversarial access are most similar to the setup in data poisoning attacks. In this setting, adversaries also seek to influence the behavior at test time on unmodified inputs by tampering solely with the training set of a model. Data poisoning attacks can be differentiated along two axes: the adversary’s intention for test-time errors and the nature of training set modifications the adversary is allowed to make. First, adversaries can seek to induce

either targeted errors (e.g., they wish for a particular image to be classified in a certain way), untargeted (they wish that a particular image is misclassified without a particular direction of the error), or indiscriminate (they wish for the trained model to not perform well on any examples). Second, adversaries can modify the training dataset by adding poisoned samples, by perturbing a subset of existing training samples, or by modifying the labels of training set examples. An example of a targeted attack that modifies the labels of certain training points in order to induce errors in Support Vector Machine (SVM) classifiers was developed in [12]. The first work to use the term “poisoning” in referring to its attacks is a targeted attack that adds malicious data points to the training set that shift the decision boundary of an SVM classifier [13].

Of most interest to us are targeted attacks that perturb existing training set examples. In particular, a subset of the recent literature studying poisoning attacks against deep learning models has focused on “clean-label” perturbations. “Clean-label” perturbations do not alter the semantics of the poisoned points and, hence, are labeled correctly either when published or by the entity performing model training. The first work to tackle this problem is [120] and subsequent improvements were developed in [156] and [3]. All three of these works focus on a restricted setting: transfer learning based on a publicly available feature extractor. In other words, the poisoning victim is not assumed to perform training from scratch on the poisoned dataset but rather downloads and reuses a publicly available feature extractor (e.g., a network trained on the ImageNet dataset [29] with its last layer replaced for a specific task).

Within the targeted setting, other works have demonstrated more powerful methods that can influence the behavior of a network even when it is trained from scratch: [97] use back-gradient optimization, [98] use generative adversarial nets, [61] simulate and unroll the training procedure and compute poisoning perturbations based on that, and [41] align the gradients of poisoned examples with those of a target inference example.

Work relying on second-order adversarial gradients to disrupt training appears in [37]. We carry out experiments that compare the effectiveness of the approach to ours in Chapter 5. A major distinction between our work and this one is that we rely on hand-crafted approaches and do not use gradient information.

Several other data poisoning methods bear discussion for completeness. A popular category of approaches to poisoning involves computing so-called influence functions that measure which training examples and which features of theirs contributed to a classification of an inference example [36, 73]. However, such approaches have been found to be fragile for state-of-the-art deep models used in computer vision [10] and they require knowledge of the exact inference sample that should be disrupted, so they do not allow for indiscriminate disruption. Next, TensorClog [125] is a method developed to artificially induce the vanishing gradient problem by poisoning the training dataset; however, this approach has had limited success and also assumes knowledge of a fixed feature extractor. Finally, [66] and [128] demonstrate that the fairness of models at test time can be degraded when the training set is poisoned. However, [128] only works with SVM models (which enable a different kind of poisoning algorithms not applicable to deep vision models) and the method from [66] is meant to do only targeted damage to predictions on a subset of the test set while retaining performance on the full one.

## ***2.6 Biases and Other Causes of Failure in Training Models***

Convolutional neural networks do not fail just in adversarial scenarios. Indeed, the machine learning and computer vision research communities have long been exploring “natural” causes of failure. Since those indicate a different set of weaknesses in the current machine learning pipeline that we may wish to exploit, we discuss several important works that observe how training can fail to yield robust and well-generalizing models.

To begin with, a well-observed finding from adversarial machine learning is that retraining models on adversarial examples returns models with decreased accuracy on a “clean” test set. This was first observed in [88]. More recently, [65] conjectured that this phenomenon can be explained by the presence of “non-robust” features in natural training sets. Those refer to features that do not align with human understanding but help models generalize better to the test distribution. Experiments from that work show that models trained solely on non-robust features of the original training set images can achieve good accuracy on the standard test sets, indicating that these features do have a substantial contribution to the good performance of models trained in the standard way. Since



models trained on adversarial examples cannot make use of these non-robust features, they do not perform as well. Similar observations have been made on high-frequency features in images (that humans cannot see): neural networks seem to learn to use correlations in that part of the feature space to make proper predictions, according to [71] and [147]. This may even help explain why neural networks are capable of fitting an entirely random dataset with no meaningful semantic correlations between the data and its labels (as first observed in [154]).

Even with the realm of semantic features, neural network models are known to fit correlations that are present in the training data but may not be true predictors of an image's class. For example, [42] claim that neural networks are biased towards using texture (e.g., the fur of a cat versus the scales on a crocodile) over shape information (e.g., a cat's ears and whiskers over a crocodile's snout) in their predictions. Thus, an image of a cat with the texture of a crocodile would be more likely to be classified as a crocodile without training on specially-designed datasets to account for that bias. Another well-known result is that convolutional classifiers may prefer the background in making predictions over the object of interest: for example, if sheep appear in a tree, they are more likely to be classified as birds [122]. In more realistic settings, cows that appear on sandy backgrounds are more likely to be classified as camels and camels with green backgrounds are more likely to be classified as cows. Work in [99] attempts to explain this occurrence through the lens of out-of-distribution generalization and failures of empirical risk minimization (ERM) and attributes it to geometric and statistical skews in the training data. Interestingly, they create datasets yielding models with degraded test-time performance by introducing these exact skews: for example, inserting colored lines in the training set spuriously correlated with the training label induces a 10% accuracy drop on the test set. Cases where neural networks prefer to fit simpler but spurious features correlated with the label were also recently observed in the wild; [26] show that chest radiography image classifiers fail for similar reasons. In Chapter 5, we leverage observations from these works to create our own version of datasets with malicious correlations between the training examples and the true label that cause more severe degradations in test performance. We seek to create malicious features that induce even stronger test-time failures than those observed natural failings. It is also worth mentioning that learning methods have been proposed to deal with

natural generalization failures [6, 50].

## **2.7 *Miscellaneous and Orthogonal Work on Dataset Security and Privacy***

A problem orthogonal to the one explored in Chapter 5 has long been a subject of interest to the machine learning and privacy research communities: how we can enable the learning of useful models or statistical inference from datasets without revealing the raw data. Differential privacy is an approach that was originally developed to enable statistical conclusions about individual data records without revealing the exact values of all fields [32] and it has also been adapted for training deep models [2]. Additionally, InstaHide [62] provides an alternative method for releasing datasets that are “learnable” by an ML model but appear gibberish to humans. Deficiencies in that approach were pointed out in [18], which demonstrates that raw images protected with InstaHide can be recovered with relatively little computational resources. Finally, there is ongoing work on federated learning [75, 90] and homomorphic encryption [47] that allow computation of machine learning models in a decentralized way or only by revealing encrypted versions of the dataset to the party performing training. The goals of Chapter 5 differ from that line of work in that we aim to make training sets that are usable by humans but cannot be used for training machine learning models.

Since before the rise of machine learning in computer vision, image watermarking has been used to enforce ownership of visual data. Two useful surveys in this area are [127] and [109]. Of particular relevance to the goals of Chapter 5 are techniques for *visible* watermarking: [27] studied methods to remove such watermarks and make more robust ones with deep learning methods.

We also point out that a different version of the goals in Chapter 5 might be to make it detectable that a machine learning model has used a particular piece of data in its training procedure. To that end, [114] introduce the concept of “radioactive data;” when data of this nature is used to train a model, this fact is detectable and provable.

## Chapter 3

# ROBUST PHYSICAL-WORLD ATTACKS ON DEEP LEARNING VISUAL CLASSIFICATION

### 3.1 Introduction

The threat adversarial examples pose to systems using deep learning-based computer vision has gained recent attention, and previous work has made great progress in understanding the space of adversarial examples, beginning in the digital domain (e.g. by modifying images corresponding to a scene) [46, 94, 100, 137], and more recently in the physical domain [7, 9, 78, 123]. Along similar lines, our work contributes to the understanding of adversarial examples when perturbations are physically added to the *objects themselves*. We choose road sign classification as our target domain for several reasons: (1) The relative visual simplicity of road signs make it challenging to hide perturbations. (2) Road signs exist in a noisy unconstrained environment with changing physical conditions such as the distance and angle of the viewing camera, implying that physical adversarial perturbations should be robust against considerable environmental instability. (3) Road signs play an important role in transportation safety. (4) A reasonable threat model for transportation is that an attacker might not have control over a vehicle's systems, but is able to modify the objects in the physical world that a vehicle might depend on to make crucial safety decisions.

The main challenge with generating robust physical perturbations is environmental variability. Cyber-physical systems operate in noisy physical environments that can destroy perturbations created using current digital-only algorithms [87]. For our chosen application area, the most dynamic environmental change is the distance and angle of the viewing camera. Additionally, other practicality challenges exist: (1) Perturbations in the digital world can be so small in magnitude that it is likely that a camera will not be able to perceive them due to sensor imperfections. (2) Current algorithms produce perturbations that occupy the background imagery of an object. It is

extremely difficult to create a robust attack with background modifications because a real object can have varying backgrounds depending on the viewpoint. (3) The fabrication process (e.g., printing of perturbations) is imperfect.

Informed by the challenges above, we design *Robust Physical Perturbations* ( $RP_2$ ), which can generate perturbations robust to widely changing distances and angles of the viewing camera.  $RP_2$  creates a visible, but inconspicuous perturbation that only perturbs the object (e.g. a road sign) and not the object’s environment. To create robust perturbations, the algorithm draws samples from a distribution that models physical dynamics (e.g. varying distances and angles) using experimental data and synthetic transformations (Figure 3.2).

Using the proposed algorithm, we evaluate the effectiveness of perturbations on physical objects, and show that adversaries can physically modify objects using low-cost techniques to reliably cause classification errors in CNN-based classifiers under widely varying distances and angles. For example, our attacks cause a classifier to interpret a subtly-modified physical Stop sign as a Speed Limit 45 sign. Specifically, our final form of perturbation is a set of black and white stickers that an adversary can attach to a physical road sign (Stop sign). We designed our perturbations to resemble graffiti, a relatively common form of vandalism. It is common to see road signs with random graffiti or color alterations in the real world as shown in Figure 3.1 (the left image is of a real sign in a city). If these random patterns were adversarial perturbations (right side of Figure 3.1 shows our example perturbation), they could lead to severe consequences for autonomous driving systems, without arousing suspicion in human operators.



Figure 3.1: The left image shows real graffiti on a Stop sign, something that most humans would not think is suspicious. The right image shows our a physical perturbation applied to a Stop sign. We design our perturbations to mimic graffiti, and thus “hide in the human psyche.”

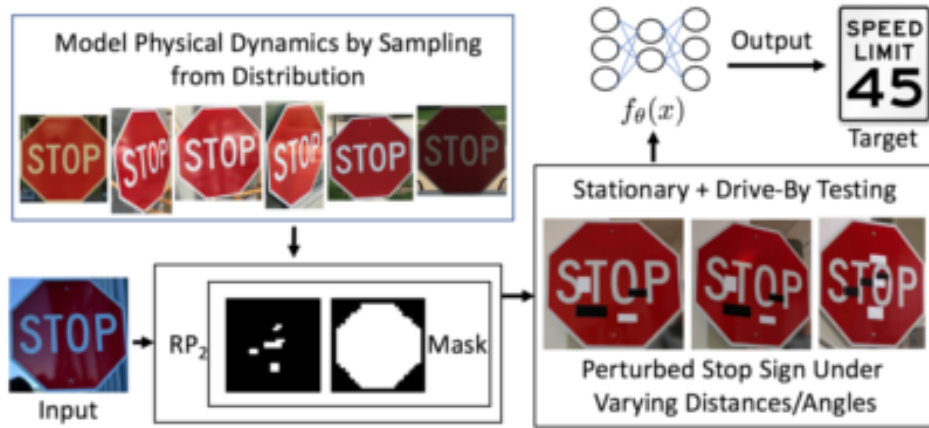


Figure 3.2:  $RP_2$  pipeline overview. The input is the target Stop sign.  $RP_2$  samples from a distribution that models physical dynamics (in this case, varying distances and angles), and uses a mask to project computed perturbations to a shape that resembles graffiti. The adversary prints out the resulting perturbations and sticks them to the target Stop sign.

Given the lack of a standardized method for evaluating physical attacks, we draw on standard techniques from the physical sciences and propose a two-stage experiment design: (1) A lab test where the viewing camera is kept at various distance/angle configurations; and (2) A field test where we drive a car towards an intersection in uncontrolled conditions to simulate an autonomous vehicle. We test our attack algorithm using this evaluation pipeline and find that the perturbations are robust to a variety of distances and angles.

### 3.2 Adversarial Examples for Physical Objects

Our goal is to examine whether it is possible to create robust physical perturbations for real-world objects that mislead classifiers to make incorrect predictions even when images are taken in a range of varying physical conditions. We first present an analysis of environmental conditions that physical learning systems might encounter, and then present our algorithm to generate physical adversarial perturbations taking these challenges into account.

### 3.2.1 *Physical World Challenges*

Physical attacks on an object must be able to survive changing conditions and remain effective at fooling the classifier. We structure our discussion of these conditions around the chosen example of road sign classification, which could be potentially applied in autonomous vehicles and other safety sensitive domains. A subset of these conditions can also be applied to other types of physical learning systems such as drones, and robots.

**Environmental Conditions.** The distance and angle of a camera in an autonomous vehicle with respect to a road sign varies continuously. The resulting images that are fed into a classifier are taken at different distances and angles. Therefore, any perturbation that an attacker physically adds to a road sign must be able to survive these transformations of the image. Other environmental factors include changes in lighting/weather conditions, and the presence of debris on the camera or on the road sign.

**Spatial Constraints.** Previous algorithms focusing on digital images add adversarial perturbations to all parts of the image, including background imagery. However, for a physical road sign, the attacker cannot manipulate background imagery. Furthermore, the attacker cannot count on there being a fixed background imagery as it will change depending on the distance and angle of the viewing camera.

**Physical Limits on Imperceptibility.** An attractive feature of current adversarial deep learning algorithms is that their perturbations to a digital image are often so small in magnitude that they are almost imperceptible to the casual observer. However, when transferring such minute perturbations to the real world, we must ensure that a camera is able to perceive the perturbations. Therefore, there are physical limits on how imperceptible perturbations can be, and is dependent on the sensing hardware.

**Fabrication Error.** To fabricate the computed perturbation, all perturbation values must be valid colors that can be reproduced in the real world. Furthermore, even if a fabrication device, such as a printer, can produce certain colors, there will be some reproduction error [123].

In order to successfully physically attack deep learning classifiers, an attacker should account for

the above categories of physical world variations that can reduce the effectiveness of perturbations.

### 3.2.2 Robust Physical Perturbation

We derive our algorithm starting with the optimization method that generates a perturbation for a single image  $x$ , without considering other physical conditions; then, we describe how to update the algorithm taking the physical challenges above into account. This single-image optimization problem searches for perturbation  $\delta$  to be added to the input  $x$ , such that the perturbed instance  $x' = x + \delta$  is misclassified by the target classifier  $f_\theta(\cdot)$ :

$$\min H(x + \delta, x), \quad \text{s.t.} \quad f_\theta(x + \delta) = y^*$$

where  $H$  is a chosen distance function, and  $y^*$  is the target class.<sup>1</sup> To solve the above constrained optimization problem efficiently, we reformulate it in the Lagrangian-relaxed form similar to prior work [20, 83].

$$\operatorname{argmin}_\delta \lambda \|\delta\|_p + J(f_\theta(x + \delta), y^*) \quad (3.1)$$

Here  $J(\cdot, \cdot)$  is the loss function, which measures the difference between the model’s prediction and the target label  $y^*$ .  $\lambda$  is a hyper-parameter that controls the regularization of the distortion. We specify the distance function  $H$  as  $\|\delta\|_p$ , denoting the  $\ell_p$  norm of  $\delta$ .

Next, we will discuss how the objective function can be modified to account for the *environmental conditions*. We model the distribution of images containing object  $o$  under both physical and digital transformations  $X^V$ . We sample different instances  $x_i$  drawn from  $X^V$ . A physical perturbation can only be added to a specific object  $o$  within  $x_i$ . In the example of road sign classification,  $o$  is the stop sign that we target to manipulate. Given images taken in the physical world, we need to make sure that a single perturbation  $\delta$ , which is added to  $o$ , can fool the classifier under different physical conditions. Concurrent work [9] only applies a set of transformation functions to synthetically sample such a distribution. However, modeling physical phenomena is complex and such synthetic transformations may miss physical effects. Therefore, to better capture the effects of changing

---

<sup>1</sup>For untargeted attacks, we can modify the objective function to maximize the distance between the model prediction and the true class. We focus on targeted attacks in the rest of this chapter.

physical conditions, we sample instance  $x_i$  from  $X^V$  by both generating experimental data that contains actual physical condition variability as well as synthetic transformations. For road sign physical conditions, this involves taking images of road signs under various conditions, such as changing distances, angles, and lighting. This approach aims to approximate physical world dynamics more closely. For synthetic variations, we randomly crop the object within the image, change the brightness, and add spatial transformations to simulate other possible conditions.

To ensure that the perturbations are only applied to the surface area of the target object  $o$  (considering the *spatial constraints* and *physical limits on imperceptibility*), we introduce a mask. This mask serves to project the computed perturbations to a physical region on the surface of the object (i.e. road sign). In addition to providing spatial locality, the mask also helps generate perturbations that are visible but inconspicuous to human observers. To do this, an attacker can shape the mask to look like graffiti—commonplace vandalism on the street that most humans expect and ignore, therefore hiding the perturbations “in the human psyche.” Formally, the perturbation mask is a matrix  $M_x$  whose dimensions are the same as the size of input to the road sign classifier.  $M_x$  contains zeroes in regions where no perturbation is added, and ones in regions where the perturbation is added during optimization.

In the course of our experiments, we empirically observed that the position of the mask has an impact on the effectiveness of an attack. We therefore hypothesize that objects have strong and weak physical features from a classification perspective, and we position masks to attack the weak areas. Specifically, we use the following pipeline to discover mask positions: (1) Compute perturbations using the  $L_1$  regularization and with a mask that occupies the entire surface area of the sign.  $L_1$  makes the optimizer favor a sparse perturbation vector, therefore concentrating the perturbations on regions that are most vulnerable. Visualizing the resulting perturbation provides guidance on mask placement. (2) Recompute perturbations using  $L_2$  with a mask positioned on the vulnerable regions identified from the earlier step.

To account for *fabrication error*, we add an additional term to our objective function that models printer color reproduction errors. This term is based upon the Non-Printability Score (NPS) by Sharif et al. [123].



Given a set of printable colors (RGB triples)  $P$  and a set  $R(\delta)$  of (unique) RGB triples used in the perturbation that need to be printed out in physical world, the non-printability score is given by:

$$NPS(\delta) = \sum_{\hat{p} \in R(\delta)} \prod_{p' \in P} |\hat{p} - p'| \quad (3.2)$$

Based on the above discussion, our final robust spatially-constrained perturbation is thus optimized as:

$$\begin{aligned} \operatorname{argmin}_{\delta} \lambda \|M_x \cdot \delta\|_p + NPS \\ + \mathbb{E}_{x_i \sim X^V} J(f_{\theta}(x_i + T_i(M_x \cdot \delta)), y^*) \end{aligned} \quad (3.3)$$

Here we use function  $T_i(\cdot)$  to denote the alignment function that maps transformations on the object to transformations on the perturbation (e.g. if the object is rotated, the perturbation is rotated as well).

Finally, an attacker will print out the optimization result on paper, cut out the perturbation ( $M_x$ ), and put it onto the target object  $o$ . As our experiments demonstrate in the next section, this kind of perturbation fools the classifier in a variety of viewpoints.<sup>2</sup>

### 3.3 Experiments

In this section, we empirically evaluate the proposed algorithm  $RP_2$ . We first evaluate a safety sensitive example, Stop sign recognition, to demonstrate the robustness of the proposed physical perturbation. To demonstrate the generality of our approach, we then attack Inception-v3 to misclassify a microwave as a phone.

While results are discussed here, detailed tables and figures are only given in Appendix A in order to aid with readability.

#### 3.3.1 Dataset and Classifiers

We built two classifiers based on a standard crop-resize-then-classify pipeline for road sign classification as described in [106, 119]. Our LISA-CNN uses LISA, a U.S. traffic sign dataset containing

---

<sup>2</sup>For our attacks, we use the ADAM optimizer with the following parameters:  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 10^{-8}$ ,  $\eta \in [10^{-4}, 10^0]$

47 different road signs [93]. However, the dataset is not well-balanced, resulting in large disparities in representation for different signs. To alleviate this problem, we chose the 17 most common signs based on the number of training examples. LISA-CNN’s architecture is defined in the Cleverhans library [104] and consists of three convolutional layers and an FC layer. It has an accuracy of 91% on the test set.

Our second classifier is GTSRB-CNN, that is trained on the German Traffic Sign Recognition Benchmark (GTSRB) [130]. We use a publicly available implementation [152] of a multi-scale CNN architecture that has been known to perform well on road sign recognition [119]. Because we did not have access to German Stop signs for our physical experiments, we replaced the German Stop signs in the training, validation, and test sets of GTSRB with the U.S. Stop sign images in LISA. GTSRB-CNN achieves 95.7% accuracy on the test set. When evaluating GTSRB-CNN on our own 181 stop sign images, it achieves 99.4% accuracy.

### 3.3.2 *Experimental Design*

To the best of our knowledge, there is currently no standardized methodology for evaluating physical adversarial perturbations. Based on our discussion from Section 3.2.1, we focus on angles and distances because they are the most rapidly changing elements for our use case. A camera in a vehicle approaching a sign will take a series of images at regular intervals. These images will be taken at different angles and distances, therefore changing the amount of detail present in any given image. Any successful physical perturbation must cause targeted misclassification in a range of distances and angles because a vehicle will likely perform voting on a set of frames (images) from a video before issuing a controller action. Our current experiments do not explicitly control ambient light, and as is evident from experimental data (Section 3.3), lighting varied from indoor lighting to outdoor lighting.

Drawing on standard practice in the physical sciences, our experimental design encapsulates the above physical factors into a two-stage evaluation consisting of controlled lab tests and field tests.

**Stationary (Lab) Tests.** This involves classifying images of objects from stationary, fixed positions.

1. Obtain a set of clean images  $C$  and a set of adversarially perturbed images ( $\{\mathcal{A}(c)\}, \forall c \in C$ ) at varying distances  $d \in D$ , and varying angles  $g \in G$ . We use  $c^{d,g}$  here to denote the image taken from distance  $d$  and angle  $g$ . The camera’s vertical elevation should be kept approximately constant. Changes in the camera angle relative the the sign will normally occur when the car is turning, changing lanes, or following a curved road.
2. Compute the attack success rate of the physical perturbation using the following formula:

$$\frac{\sum_{c \in C} \mathbb{1}_{\{f_{\theta}(\mathcal{A}(c^{d,g}))=y^* \wedge f_{\theta}(c^{d,g})=y\}}}{\sum_{c \in C} \mathbb{1}_{\{f_{\theta}(c^{d,g})=y\}}} \quad (3.4)$$

where  $d$  and  $g$  denote the camera distance and angle for the image,  $y$  is the ground truth, and  $y^*$  is the targeted attacking class.<sup>3</sup>

Note that an image  $\mathcal{A}(c)$  that causes misclassification is considered as a successful attack only if the original image  $c$  with the same camera distance and angle is correctly classified, which ensures that the misclassification is caused by the added perturbation instead of other factors.

**Drive-By (Field) Tests.** We place a camera on a moving platform, and obtain data at realistic driving speeds. For our experiments, we use a smartphone camera mounted on a car.

1. Begin recording video at approximately 250 ft away from the sign. Our driving track was straight without curves. Drive toward the sign at normal driving speeds and stop recording once the vehicle passes the sign. In our experiments, our speed varied between 0 mph and 20 mph. This simulates a human driver approaching a sign in a large city.
2. Perform video recording as above for a “clean” sign and for a sign with perturbations applied, and then apply similar formula as Eq. 3.4 to calculate the attack success rate, where  $C$  here represents the sampled frames.

---

<sup>3</sup>For untargeted adversarial perturbations, change  $f_{\theta}(e^{d,g}) = y^*$  to  $f_{\theta}(e^{d,g}) \neq y$ .

An autonomous vehicle will likely not run classification on every frame due to performance constraints, but rather, would classify every  $j$ -th frame, and then perform simple majority voting. Hence, an open question is to determine whether the choice of frame ( $j$ ) affects attack accuracy. In our experiments, we use  $j = 10$ . We also tried  $j = 15$  and did not observe any significant change in the attack success rates. If both types of tests produce high success rates, the attack is likely to be successful in commonly experienced physical conditions for cars.

### 3.3.3 Results for LISA-CNN

We evaluate the effectiveness of our algorithm by generating three types of adversarial examples on LISA-CNN (91% accuracy on test-set). For all types, we observe high attack success rates with high confidence. Table 3.1 summarizes a sampling of stationary attack images. In all testing conditions, our baseline of unperturbed road signs achieves a 100% classification rate into the true class.

**Object-Constrained Poster-Printing Attacks.** This involves reproducing the attack of Kurakin et al. [78]. The crucial difference is that in our attack, the perturbations are confined to the surface area of the sign excluding the background, and are robust against large angle and distance variations. The Stop sign is misclassified into the attack’s target class of Speed Limit 45 in 100% of the images taken according to our evaluation methodology. The average confidence of predicting the manipulated sign as the target class is 80.51% (second column of Table A.1).

For the Right Turn warning sign, we choose a mask that covers only the arrow since we intend to generate subtle perturbations. In order to achieve this goal, we increase the regularization parameter  $\lambda$  in equation (3.3) to demonstrate small magnitude perturbations. We achieve a 73.33% targeted-attack success rate (Table 3.1). Out of 15 distance/angle configurations, four instances were not classified into the target. However, they were still misclassified into other classes that were not the true label (Yield, Added Lane). Three of these four instances were an Added Lane sign—a different type of warning. We hypothesize that given the similar appearance of warning signs, small perturbations are sufficient to confuse the classifier.

**Sticker Attacks.** Next, we demonstrate how effective it is to generate physical perturbations in

the form of stickers, by constraining the modifications to a region resembling graffiti or art. The fourth and fifth columns of Table 3.1 show a sample of images, and Table A.1 (columns 4 and 6) shows detailed success rates with confidences. In the stationary setting, we achieve a 66.67% targeted-attack success rate for the graffiti sticker attack and a 100% targeted-attack success rate for the sticker camouflage art attack. Some region mismatches may lead to the lower performance of the LOVE-HATE graffiti.

**Drive-By Testing.** Per our evaluation methodology, we conduct drive-by testing for the perturbation of a Stop sign. In our baseline test we record two consecutive videos of a clean Stop sign from a moving vehicle, perform frame grabs at  $k = 10$ , and crop the sign. We observe that the Stop sign is correctly classified in all frames. We similarly test subtle and abstract art perturbations for LISA-CNN using  $k = 10$ . Our attack achieves a targeted-attack success rate of 100% for the subtle poster attack, and a targeted-attack success rate of 84.8% for the camouflage abstract art attack. See the supplemental materials for sample frames from the drive-by video.

### 3.3.4 Results for GTSRB-CNN

To show the versatility of our attack algorithms, we create and test attacks for GTSRB-CNN (95.7% accuracy on test-set). Based on our high success rates with the camouflage-art attacks, we create similar abstract art sticker perturbations. The last column of Table 3.1 shows a subset of experimental images. Table A.4 summarizes our attack results—our attack fools the classifier into believing that a Stop sign is a Speed Limit 80 sign in 80% of the stationary testing conditions. Per our evaluation methodology, we also conduct a drive-by test ( $k = 10$ , two consecutive video recordings). The attack fools the classifier 87.5% of the time.

### 3.3.5 Results for Inception-v3

To demonstrate generality of  $RP_2$ , we computed physical perturbations for the standard Inception-v3 classifier [77, 135] using two different objects, a microwave and a coffee mug. For the microwave, our adversarial sticker causes the classifier to misclassify it as our target class, “phone,” in 90% of

the tests (Table A.5). For the coffee mug, our adversarial sticker causes the classifier to misclassify it as our target class, “cash machine”, in 71.4% of the tests. Figure A.7 shows an example of the adversarial sticker for microwave and Table A.8 presents examples for the mug.






















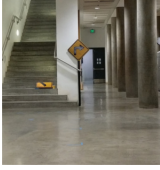


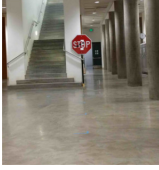
Note that for both attacks, we have reduced the range of distances used due to the smaller size of the cup and microwave compared to a road sign (e.g. Coffee Mug height: 11.2cm, Microwave height: 24cm, Right Turn sign height: 45cm, Stop Sign: 76cm). Table A.5 summarizes our attack results on the microwave and Table A.6 summarizes our attack results on the coffee mug. For the microwave, the targeted attack success rate is 90%. For the coffee mug, the targeted attack success rate is 71.4% and the untargeted success rate is 100%.

### 3.4 Discussion

**Black-Box Attacks.** Given access to the target classifier’s network architecture and model weights,  $RP_2$  can generate a variety of robust physical perturbations that fool the classifier. Through studying a white-box attack like  $RP_2$ , we can analyze the requirements for a successful attack using the strongest attacker model and better inform future defenses. Evaluating  $RP_2$  in a black-box setting is an open question.

**Image Cropping and Attacking Detectors.** When evaluating  $RP_2$ , we manually controlled the cropping of each image every time before classification. This was done so the adversarial images would match the clean sign images provided to  $RP_2$ . Later, we evaluated the camouflage art attack using a pseudo-random crop with the guarantee that at least most of the sign was in the image. Against LISA-CNN, we observed an average targeted attack rate of 70% and untargeted attack rate of 90%. Against GTSRB-CNN, we observed an average targeted attack rate of 60% and untargeted attack rate of 100%. We include the untargeted attack success rates because causing the classifier to not output the correct traffic sign label is still a safety risk. Although image cropping has some effect on the targeted attack success rate, our recent work shows that an improved version of  $RP_2$  can successfully attack object detectors, where cropping is not needed [129].

Table 3.1: Sample of physical adversarial examples against LISA-CNN and GTSRB-CNN.

Dist./Angle	Subtle Poster	Subtle Poster Right Turn	Camouflage Graffiti	Camouflage Art (LISA- CNN)	Camouflage Art (GTSRB- CNN)
5' 0°					
5' 15°					
10' 0°					
10' 30°					
40' 0°					
Targeted- Attack	100%	73.33%	66.67%	100%	80%
Success					

## Chapter 4

### **FOGGYSIGHT: A SCHEME FOR FACIAL LOOKUP PRIVACY**

#### **4.1 Introduction**

Unfortunately, CNNs are not always applied with good intentions in mind. The progress in face recognition in particular has also enabled unprecedented invasions of individual privacy. At least two services are currently being pioneered that offer face search databases for law enforcement agencies and even for the broader public [51, 55, 57]. It is easy to imagine nefarious applications of this easily accessible technology. Stalkers, who have only seen potential victims online, could apply this technology to identify individuals in public web cam video streams; see [131] for a motivating example. Criminal or other illegal organizations could also use this technology to identify people in news media photos and then target those people for physical harm or retaliation; see [103] for a motivating example. These examples illustrate that individuals uploading pictures to social media websites are exposing themselves to the risk of future identification in new photos via DL-enabled facial searches.

Any solution to protect individual privacy must acknowledge these realities: that facial search databases already contain previously publicly available tagged photos of many (possibly millions of) individuals, that individuals cannot predict when they are at risk of being targeted by a face recognition system, and those photos being used for face recognition may come from sources external to the social media platforms. In this chapter, we propose a new framework for protecting against face recognition that takes these issues into account. We propose using adversarial examples – small perturbations to images that fool DL models but are imperceptible to humans – to poison the lookup database of facial search services. This involves coordination of adversarial modifications among many users: a large number of adversarial photos uploaded by many different individuals may protect privacy by “crowding out” previously scraped “clean” photos of individuals in response



to queries *without* those individuals needing to obscure their identity when in public.

## 4.2 Definition of Terms

To aid with further discussion of face recognition, we introduce several terms and notations. We denote the face recognition model as  $f : \mathbb{R}^{w \times h \times c} \mapsto \mathbb{R}^d$  for some latent embedding space of dimension  $d$  and images of size  $w \times h \times c$ . In this embedding space, similarity between two faces is computed using normalized distance in the embedding space. That is, for two images  $x_1, x_2$ , the distance function  $D$  between them is evaluated as:

$$D(x_1, x_2) = \left\| \left\| \frac{f(x_1)}{\|f(x_1)\|_2} - \frac{f(x_2)}{\|f(x_2)\|_2} \right\|_2 \right\|_2^2$$

In addition, we define the following terms:

- **Lookup Set:** The set of photos that a face recognition company scrapes from social media. These photos, along with their associated profiles or links, are those that are cross-referenced against when identifying an individual in a new photo. Each photo in the lookup set represents an embedding vector in the neural network’s output space – the nearest lookup set photos to the query photo are returned when performing a search. We denote the lookup set by  $L$ .
- **Query Photo:** The photo that the user of face recognition technology (the adversary in our model, see Section 4.3) wants to match to an identity. This photo may be a photo of, for example, a stranger in a public place. This photo is processed by the neural network and a vector is produced in the embedding space that can be compared against the vectors of the lookup set photos. The closest neighbors of the lookup set photos in embedding space are returned as candidate matches.
- **Top  $k$  Recall Set** The set of  $k$  closest neighbors in the lookup set to the embedding vector corresponding to the query photo.  $k$  is a parameter that can be adapted for broader or narrower searches. For some query photo  $q$ , lookup set  $L$  and distance metric  $D$ , we use the following

notation to denote this set:

$$N(q, k) := \arg \text{top-k}_{x \in L}(D(q, x))$$

Some facial recognition services – such as the Microsoft Azure Face Service that we study – only expose  $N(q, 1)$  in their API responses.

### 4.3 Goals and Assumptions

Our objective is to prevent previously scraped public photos of social media users from being useful to facial search services by poisoning the database of facial images.

In striving for this goal, we make the following assumptions, which we believe correspond to the real-world deployment scenario of such face recognition services.

**The Face Recognition Service is the Adversary.** We treat the company supplying face recognition technology — one that scrapes public photos of users from various social media websites — as the adversary. The source of such scraped photos may include such services as Facebook, Twitter, LinkedIn, Venmo, and others. This adversary records which account each photo came from or who was tagged so that the lookup photos can be linked to identities or the account that the photo was tagged for, if available. When a third party performs a face lookup through the adversary, the system processes the query photo with a neural network, computes the photo’s closest neighbors, and returns the accounts associated with those. In this model, we do not restrict how the third party obtains the query photo; it could be an untagged photo from a different social media company (one not scraped by the adversary), from a surveillance camera photo, or from other sources.

Importantly, we assume this scraping has already taken place for millions of users and is ongoing for others and for future photos of those already in the database. Our solution seeks to improve privacy, given that the adversary possesses some fixed amount of unmodified photos associated with individuals and that the adversary will only pick up modified photos of individuals participating in our scheme once our solution is fully deployed.

**Social Media Users Seek Privacy.** Users of the platforms enumerated above seek to frustrate the search by ensuring that links to their profiles are not returned when the query photo truly is of them.

Where that is not possible, users prefer that many other identities are returned by the search so that theirs does not stand out. Users may collaborate to achieve this goal and the platforms hosting the photos might also participate in the privacy enhancing scheme. We discuss different collaboration models in section 4.4.

**No User Control over the Query Photo.** Crucially, we assume that individuals do not control the query photos that malicious parties might submit to the face recognition service to identify them. Individuals may not be in full control of their appearance whenever photos of them might be taken in public. In addition, they might not wish to permanently modify their physical appearance whenever they are in public spaces, but might be willing to participate in a scheme such as ours that involves digital modifications that do not lower the quality of their digital photos.

**Limited Control over the Lookup Set.** Individuals have the ability to control future photos that get scraped by the facial lookup service because they control the photos they upload to social media. It is useful to distinguish between two types of individuals here. One is individuals who do not have photos already scraped by the adversary. This might be because all of their photos were private or because they never uploaded any photos in the first place. Another is individuals who already have images in the adversary’s database. These individuals can begin participating in our privacy protection scheme and modify their future uploads, which the adversary then scrapes. However, they cannot modify the photos that were previously scraped. Thus, the adversary possesses a “core” set of clean images for those individuals. Untagging, delisting and otherwise hiding previously scraped photos is unlikely to be an effective protection for these people, as the links between their images and their profiles already exist in the adversary’s database. Our solution aims to increase the privacy of this second group of individuals.

**Access to the Model.** We assume that the protectors have access to the adversary’s model and weights so that they can perform so-called “white-box” adversarial examples modifications. This assumption is not unreasonable by itself, as models often leak even from highly secure organizations. In this particular scenario, the adversary may even be forced by regulators to release the model publicly for accountability and transparency purposes. It is also possible that the adversary is

outright using a public face recognition model that the protectors also have access to. Without such a level of access, protectors can rely on the transferability property of adversarial examples to carry out their attacks.

We study how protectors can adapt their decoy generation and the effects on our scheme’s privacy protections in Section 4.6.3.

**No Quality Degradation of Social Media Photos.** We wish to apply a privacy defense mechanism that does not degrade the quality of photos that users post. Legitimate human users should still be able to recognize people they know in modified photos. Any modifications introduced to encumber computational processing of facial images should not impede human understanding. Our system provides a tunable knob for defense, whereby tuning the knob for increased privacy can lead to more visual artifacts. The knob settings we consider in this paper are still effective for privacy, even though they introduce only minor artifacts. Although outside the scope of this paper, a user study could evaluate the visual impact of these artifacts, for large knob settings. For one such existing study, we refer readers to [39].

#### **4.4 The FoggySight Design**

As facial lookup is primarily enabled through DL algorithms, we propose using adversarial examples<sup>1</sup> for providing privacy for social media users. These are modifications to photos that shift the output of neural networks according to the modifier’s choosing. Usually, such “adversarial” changes are imperceptible to humans, making them particularly attractive tools for our use case.

While the generation of adversarial examples has been well-studied in the literature, we explore how they can be used for privacy enhancements. Thus, we explore questions around picking adversarial targets and coordination among users in doing so to achieve their privacy defense goals.

Instead of focusing on how the outputs of a model or a face recognition service are affected by individual adversarial examples, we consider the broader facial search process and optimize for

---

<sup>1</sup>Recall that for FoggySight, the adversary is the facial lookup service. The “adversarial” designation in “adversarial examples” refers to adversaries against the neural network model. In our case, the adversaries against the neural network model are the users seeking privacy from their adversary — the facial lookup service.

privacy in the recall set. That is, images associated with the true protected individual in the lookup set should not be returned when the service is queried with their photo or they should be returned only along a multitude of other identities.

In order to discuss how we generate adversarial examples, we will first introduce some notation. For specific identities  $i$  and  $j$ , we will denote the photos that depict identity  $i$  and  $j$  in the lookup set as  $L_i$  and  $L_j$  respectively. We will use  $x_i$  and  $x_j$  to denote elements of  $L_i$  and  $L_j$ , and  $q_i$  and  $q_j$  to denote query photos depicting identities  $i$  and  $j$ , respectively. With this terminology, we can summarize the face recognition pipeline as follows:

1. The face recognition company scrapes a lookup set from publicly available sources and obtains a trained network  $f$ .
2. The user of the face recognition technology takes a query photo  $q_i$  of some identity  $i$ .
3. The face recognition technology computes the top  $k$  recall set  $N(q_i, k)$  with respect to  $q_i$  and returns them to the user (i.e., the adversary in our model), along with associated links or profiles associated with those photos in  $N(q_i, k)$ .
4. The user (the adversary in our model) manually examines the set of identities in  $N(q_i, k)$  and uses their own judgment to recover the true identity of the person depicted in  $q_i$ . If many of the photos in  $N(q_i, k)$  are also in  $L_i$ , then the user will be able to match  $q_i$  to the identity  $i$ .

With this in mind, the goal of our adversarial examples is to prevent many of the photos in the set  $L_i \subset L$  from being in  $N(q_i, k)$ .

#### 4.4.1 Overview

As we discuss in Section 4.3, an individual  $i$  that cannot modify all of their photos in  $L_i$ , for example because they have already been scraped by the adversary. Clean photos in  $L_i$  will be close to future query photos  $q_i$  and will likely be contained in  $N(q_i, k)$ , thus deanonymizing the individual.

In order to protect privacy in this scenario, we propose instead to “crowd out” as many of the clean lookup set photos as possible. That is, we propose embedding as many decoy photos as possible with different identities into the embedding space near an individual’s clean lookup photos such that those decoy photos show up in future queries, rather than the lookup photos themselves. If the lookup set contains many photos of *other* identities that are closer to a future query photo than the clean lookup photos are, then the search will fail to recover who is truly depicted in the query. To better describe how this scheme operates, let us introduce several new terms:

- **Protected users:** Those are users whose identity the scheme aims to protect or hide from the facial lookup.
- **Protectors:** Those are the users who choose to volunteer photos for the crowding out effect. By volunteering these photos, they achieve minimal additional privacy for themselves (similar to the privacy benefits from the “solo action” solution). However, they contribute to the privacy of protected users. Users can be both protected and protectors but we highlight these different groups to show that the benefit is concentrated on the protected whereas the action is needed from the protectors.
- **Decoy photos:** Photos that depict the protectors in reality but for which neural networks produce embeddings in the region of the protected. We aim to ensure that decoy photos — as opposed to clean photos of the protected — are returned in the recall set in response to a query.

With these terms in mind, the scheme operates as follows:

1. Protectors create decoy photos by means of adversarial examples-generation algorithms.
2. Protectors upload those photos to their social media profiles and make them public.
3. The adversary (the facial lookup company) scrapes those decoy photos and pre-computes their embeddings, as usual for all photos.

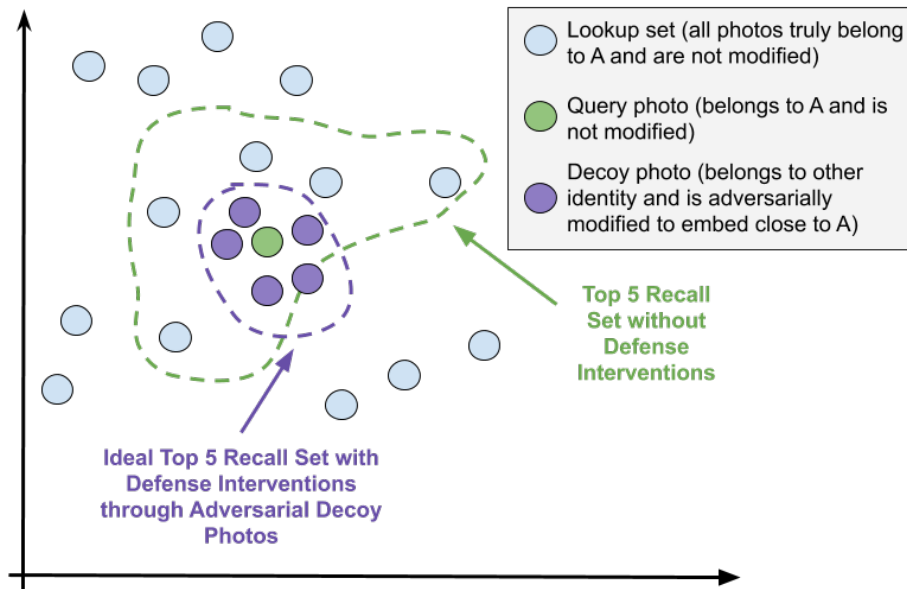


Figure 4.1: Visual illustration of the FoggySight privacy defense strategy. Decoy photos are pictures belonging to different identities that are adversarially modified so that face recognition neural networks produce embedding vectors close to those of the identity being protected (denoted as “A”). Therefore, decoy photos appear as the closest neighbors of a query photo of A and the real identity is not revealed in response to the query.

4. When a query is run on a protected identity, the closest matches are decoy photos belonging to a different identity.

For a visual representation of this idea, see Fig. 4.1.

#### 4.4.2 Adversarial Examples Generation

To generate targeted adversarial examples, given a face recognition model  $f$ , a target vector  $v \in \mathbb{R}^d$  and an image  $x \in \mathbb{R}^{w \times h \times c}$  users can solve this optimization problem for an adversarial perturbation  $\delta$ :

$$\arg \min_{\delta} D(f(x + \delta), v) \text{ such that } \|\delta\|_{\infty} \leq \epsilon .$$

This can be solved with projected gradient descent, as proposed in [88]. Note that  $\delta$  is unique to each image  $x$ . Also, note that this optimization procedure may not converge ideally and there might be a gap in the vector space between the target  $v$  and  $f(x + \delta)$ . In the next section, we discuss how pairs of  $x$  and  $v$  are to be selected for maximum effectiveness of the strategy.

In some cases, the face recognition model  $f$  that the protectors have access to may not match the model that the facial search provider uses. For those situations, the protectors can generate *robust* adversarial examples by applying the Expectations-over-Transformations (EOT) algorithm [9]. This boils down to solving the following optimization objective:

$$\arg \min_{\delta} \mathbb{E}_i [D(f(T_i(x + \delta)), v)] \text{ such that } \|\delta\|_{\infty} \leq \epsilon$$

where  $T_i$  are image transformations – such as cropping, brightness shifts, additive Gaussian noise, etc. – applied with randomly sampled parameters. In practice, we solve this objective by drawing the parameters for the transformation randomly at each step of the projected gradient descent algorithm. This boosts transferability of adversarial examples because they acquire more universal features that two different models learn to use in computing their predictions. In our experiments, we use random brightness shifts, random cropping, and additive Gaussian noise for this purpose.

A further way to boost transferability is to generate decoys against an ensemble of face recognition models (see [83]). This works by solving the following objective for models  $f_1, \dots, f_j, \dots, f_n$ :

$$\arg \min_{\delta} \mathbb{E}_i \left[ \sum_{j=1}^n D(f_j(T_i(x + \delta)), v) \right] \\ \text{such that } \|\delta\|_{\infty} \leq \epsilon$$

#### 4.4.3 Selecting Targets

For an individual with identity  $i$  and lookup photos  $L_i$ , the overall goal is to have others embed many decoy photos near the photos in  $L_i$  such that a new query photo’s neighbor set  $N(q_i, k)$  contains mostly decoy photos rather than photos from  $L_i$ . Ideally then, the targets  $v$  chosen for adversarial example generation should be embedding vectors corresponding to photos that either belong to  $L_i$ , or points close to such vectors. In this section we enumerate several different strategies for picking



such targets.

**Same Universal Target.** First, all users contributing decoy photos could select a single photo of the protected user and modify all of their images so that they embed close to that one photo. This has two benefits: simplicity and an extra layer of privacy for the defended individual. When everybody creating decoy photos has the same target, there is no problem of coordination. Everybody knows exactly how to modify their photos and does not need to check with anybody else in the scheme on what target to use. Such a mechanism also reveals the least amount of information about the protected user. This is particularly important as previous work has established that facial embedding vectors can be reversed to obtain the original appearance of the individual [38].

Unfortunately, this poisoning scheme is unlikely to be very effective. A single sample from the distribution of photos of the defended individual is probably not a good representative for the entire distribution. If the defenders are “lucky” and this is the most probable sample, then many other lookup images will be crowded out by the decoys. However, if they are not, the crowding out effect will be limited as the query photo is likely to land far away from all the decoys and closer to other clean images of the target.

**Randomly Sampled Lookup Set Photo as Target.** As a second approach, each user in the decoy-generating group could pick a random lookup set photo of the protected user as their target. The benefits of this scheme is that with large enough numbers of decoy photos, the community can easily crowd out every single lookup set photo of the user. In fact, if the run of the adversarial examples generation algorithm converges perfectly, then a linear number of decoy photo is sufficient to crowd out the clean photos, no matter where the query photo lands. This will happen because the closest photo — along with its decoys — will fill the search result set (assuming the embeddings of the decoy photos land exactly on top of the clean photos).

Unfortunately, adversarial examples algorithms do not converge perfectly in practice. Thus, to achieve perfect crowding out, the final error of the decoy photos needs to be in a favorable direction to the defenders (the decoy photos need to land between the query photo and the lookup set photos). Since neither the exact position of the query photo nor the error in the adversarial

examples generation are easily predictable, the scheme might need more than a linear number of decoys to achieve its goals. Even worse, this targeting approach requires honest cooperation by all defenders in drawing the target photos uniformly at random. Any intentional or unintentional bias in the selection of the targets (a deviation from the uniform sampling) for the decoys reduces this scheme's effectiveness.

The deficiencies of the solutions proposed above reveal that the community generating decoy photos should make use of the fact that the defenders know the structure of the lookup set a priori. They can take advantage of this fact in two ways. First, they could use the lookup set to estimate the most likely point where the query photo will land. Then, the decoy photos could be concentrated in that region. Alternatively, they could attempt to distribute the decoy photos so that they are closest to the lookup set photos that are highest likelihood. We present instantiations of these ideas next.

**Targeting the Mean Vector of the Lookup Set.** Assuming that query photos and lookup set photos are drawn from the same distribution and that it is sufficiently similar to the normal distribution, the most likely point for the query photo to land on is the mean of that distribution. This is easily estimated with the mean of the lookup set, assuming it is sufficiently large. Further, if the variance of the distribution of photos of the same identity is low, the identity photo is unlikely to be far away from the mean. Therefore, a large concentration of decoy photos around the mean should easily crowd out most lookup set photos.

**Targeting a Sample from a Fitted Distribution.** Another conjecture is that the distribution that query photos are drawn from might have higher variance than the distribution of the lookup set (but the same mean). Certainly, this is possible as query photos are likely to be sourced from uncontrolled environments that might be very different from the social media photos used to build up the lookup set (e.g., CCTV). In this situation, it is preferable to introduce decoys that do not land exactly on the mean of the lookup set. We explore drawing targets from a Gaussian distribution with mean and variance matching that of the lookup set.

#### 4.4.4 *Collaboration Models*

Regardless of the targeting strategy, protectors need to collaborate in order to achieve maximum effectiveness. We describe collaborations ranging from no collaboration at all to a fully decentralized approach with everyone participating.

**No Collaboration.** In this setting, protected users are their own protectors. They can flood Internet websites that are to be scraped or create fake accounts with decoy photos. A limitation of this approach is that a user acting alone is unlikely to be able to generate enough decoy content without violating other policies.

**Centralized Assignment.** In this setting, a trusted central party (e.g., a social media company) endeavors to protect the privacy of its own users from facial lookups. The company could apply all alterations automatically to users who opt in or to all users by default. This has the benefit that users need not coordinate or trust each other at all.

The company can make centralized decisions for targeting and adapt the scheme as necessary. The problem with this model is that the solution is not platform-agnostic and users can still be deanonymized from photos on websites that do not apply this protection.

**Decentralized Collaboration.** Users can collaborate with each other to select targets and modify their photos. This could be mediated by a browser extension or a phone app that automatically applies the needed modifications for the photos to act as decoys. Indeed, this approach does not require the consent of the protected individual at all, as protectors could even scrape the protected's public photos themselves. The downside to this approach is that coordination is difficult. Protectors may not follow the protocol correctly, they might be running outdated versions of the software, they could outright go rogue and pick arbitrary targets or not participate at all. Protectors also might not be aware if they are picking decoy photos of other protectors or if they are using the clean photos as targets.

#### 4.4.5 Matching Protectors and Protected

In all cases, the matching of protected and protectors need only follow a simple rule: No single protector should provide too many of the decoys for a given protected individual, relative to the number of decoys by other protectors. To illustrate why this rule is important, consider an extreme scenario with only one protector for a given protected individual. When a query is run for the protected individual, the single protector will appear as the most likely individual whose face belongs to that user. This means that the protector will now suffer whatever negative consequence were targeted at the protected. By contrast, if many different protectors are returned, then the facial search user will not be able to identify any individual in the query photo (mistakenly or otherwise) with any reasonable degree of certainty. This is captured by our “identity uniformity” metric (see Sections 4.5 and 4.6). Beyond this rule, FoggySight is agnostic to how protectors are matched up with protected users.

### 4.5 Experimental Setup and Metrics

In the experiments that follow, we aim to study and understand which strategy performs best in terms of protecting individual privacy. In order to do so, we need to define quantitative metrics that represent success when it comes to privacy protection.

#### 4.5.1 Metrics

The first metric we call *recall percentage at k*. Intuitively, it is defined as the percent of the target’s photos that appear in the top  $k$  matches from the lookup set. This is meant to reflect a scenario in which the user of a face recognition system has a limited ability to look through the top  $k$  matches. It is formally defined as:

$$\text{RP}_k(A, q_i) = \frac{\sum_{x \in N(q_i, k)} \mathbb{I}[x \in L_i]}{k} \quad (4.1)$$

where  $q_i$  is a query image depicting individual  $i$ ,  $L_i$  is the photos in the lookup set that also depict individual  $i$ , and  $\mathbb{I}$  denotes the indicator function. We assume that procedure  $A$  has been used to

modify some portion of the images in the lookup set  $L$ .

The second metric we call *discovery rate at  $k$* . Intuitively, it is defined as the percentage of the time that any photo from the target identity appears in the top  $k$  matches from the lookup set. This is meant to reflect the scenario in which the user of the face recognition system has the resources to look through and investigate every single photo in the top  $k$  matches. Formally, we define it as:

$$\text{DR}_k(A, q_i) = \mathbb{I}[\exists x \in N(q_i, k) \text{ such that } x \in L_i] \quad (4.2)$$

That is, it is 1 if there exists at least one photo of individual  $i$  in the neighborhood around  $q_i$ . Although the discovery rate for a single image  $x$  is either 0 or 1, we can take the expectation over many images from a single identity to get the expected discovery rate for that identity, or over all images in  $L$  to get the expected discovery rate for the adversarial procedure  $A$ .

The third metric we call *identity uniformity at  $k$* . Intuitively, it captures how many different identities are present in the recall set (subject to normalization). Lower identity uniformity (close to 0.0) means that every possible identity is included in response to a query. Thus, privacy is protected because the privacy adversary cannot be reasonably certain which identity of all the possible ones is depicted in the query (it could be any of them). Higher identity uniformity means the privacy adversary can reasonably examine all returned identities closer to violate the privacy of the person in the query. Formally, we define identity uniformity for a query photo  $q_i$  as:

$$\text{IDUnif}_k(q_i) = 1 - \frac{ID(N(q_i, k))}{ID(L)} \quad (4.3)$$

where  $ID$  is a function that maps a set of images to the number of unique identities depicted in those images. As with recall and discovery, we take the expectation over all photos serving as queries.

#### 4.5.2 Dataset and Models

In our exploration, we use the VGGFace2 test dataset [17] for evaluation. In order to make the exploration tractable given limited computational resources, we sampled 19 identities and 50 photos of each uniformly at random and performed all experiments on them. The original test dataset that we sample from has 500 identities. The full VGGFace2 dataset consists of 9,000 identities in total

with an average of 362 faces per subject. During our explorations, we additionally ran some of our experiments on the full dataset and did not find the results to be substantially different. Therefore, we believe the results that we present are more broadly applicable, despite the subsampling.

To perform our experiments, we modify each of the 50 photos of each of the 19 identities 18 times – one for each other identity – by using the algorithms and targeting schemes set out in Section 4.4.2. The 50 clean photos of each identity are also used to compute the targets as set out in Section 4.4.3. Then, we sample from the resulting decoys and from the original (subsampling) set of clean photos to build up a lookup set. This set corresponds to the poisoned dataset that the facial search system would scrape from the Internet to provide its service. Query photos are selected at random from the remaining photos (that were not included in the 50) in each identity to simulate an image that was taken of the target in public. All metrics reported are averaged over multiple query photos.

We perform all experiments on the Inception-ResNet v1 network [134] trained on the VGGFace2 training set [17] and originally implemented at the following GitHub repository: <https://github.com/nyoki-mtl/keras-facenet>, which is itself a reimplement of this repository: <https://github.com/davidsandberg/facenet>. For transferability experiments (Section 4.6.3), we use the original implementations at <https://github.com/davidsandberg/facenet> and use a second network with the same architecture but trained on the Casia-Webface dataset [153]. We do not process any images from the Casia-Webface dataset but merely use the pretrained network.

We also study transferability to the Microsoft Azure Face API service available here: <https://azure.microsoft.com/en-us/services/cognitive-services/face/>. This service allows its users to specify a training set of images associated with a set of identities. For this purpose, users create “person groups.” These person groups are loaded with images for each person and then trained, but the documentation does not provide details on what kind of model is used for this purpose. When a person group is queried, the service responds with the identity of the person it believes is in the photo or with an empty response if it does not identify anybody from the person group’s members. In our measurements, we consider only a response with the correct identity as a

correct response and empty responses and responses with an identity not matching the ground truth of the query photo are considered wrong. Thus, for experiments on the Azure Face Service, we only report the equivalent of recall at  $k = 1$ .

### 4.5.3 Implementation Details

For the results in Sections B.1, 4.6.1, and 4.6.2, we use a learning rate of  $\alpha = 0.1$  and batch size of 128, and run PGD for up to 400 iterations. We interrupt the optimization if the loss value has not declined for 10 consecutive iterations.  $\epsilon$  is set as indicated in the figures. Experiments in these sections are implemented in Tensorflow 2.0 [1] and use the network provided at <https://github.com/nyoki-mtl/keras-facenet>.

For the results in Section 4.6.3, we use  $\alpha = 0.01$  and run the PGD algorithm for 2000 iterations without early stopping. We apply the following transformations with parameters sampled at random at each gradient step: random flip left or right, random brightness shift by up to 0.25, random crop of a rectangle of size  $150 \times 150$ , with resizing to the network input size of  $160 \times 160$ , additive Gaussian noise with  $\mu = 0.0$  and  $\sigma = 0.5$ . Experiments in this section are implemented in Tensorflow 1.15 [1] and use the Inception ResNet-v4 networks implemented at <https://github.com/davidsandberg/facenet> and trained on VGGFace2 [17] and Casia-Webface [153].

In order to be able to carry out experiments in a reasonable amount of time, we have sampled 19 identities uniformly at random from the VGGFace2 test dataset. Those identities are as follows: n000958, n001683, n001781, n002503, n002647, n002763, n003215, n003356, n004658, n005303, n005359, n005427, n007548, n008613, n008655, n009114, n009232, n009288, n000029. We have further sampled 50 photos from each identity to include in our lookup sets and to serve as the basis for generating decoys. This list of 1,000 photos is too large to include in the appendix, but is available upon request. During evaluation, we sample another set of 5 photos (distinct from the 50) and use them as “query photos.” All metrics reported are averaged over each of these 5 photos for each of the 19 identities.

## 4.6 Evaluation

In this section, we report results of our experimental evaluation of FoggySight. For readability, we present some results later in Appendix B.

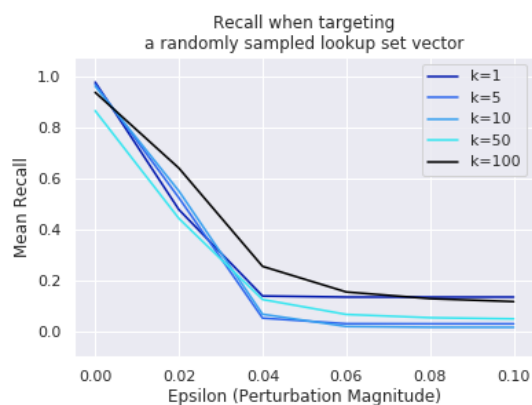
### 4.6.1 Privacy Protection Success as a Function of $\epsilon$

We analyze how well the decoys fare based on our metrics: recall, discovery, and identity uniformity. In this section, we consider two parameters: the size of the recall set  $k$  and the perturbation magnitude  $\epsilon$ . Note that  $k$  is set by the adversary whereas the protectors get to pick  $\epsilon$ . We seek to understand what  $\epsilon$  achieves the optimal tradeoff between degrading the image quality and achieving the privacy protection goals under enough various settings for  $k$ . Here, we present results with the two most effective strategies: targeting a randomly sampled lookup set photo and targeting the mean of the lookup set.

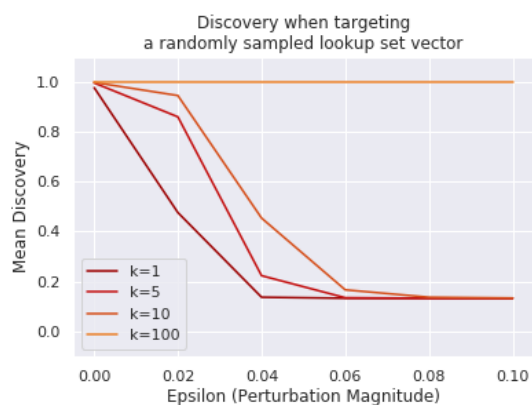
**Randomly Sampled Lookup Set Photo as Target.** We first explore using a random sample from the entire lookup set as targets for the decoy photo optimization by the protectors. The results are presented in Fig. 4.2. For  $\epsilon \geq 0.04$ , recall at  $k = 1$  is only 20%, indicating that the closest neighbor of the query belongs to the true identity less than a fifth of the time. For higher  $k$ 's, only a small percentage of the recall set ends up truly belonging to the protected identity, as can be seen by values for recall close to 0 in Fig. 4.2a. This success can also be confirmed by the low values for the discovery rate – indicating that the protected identity is present in the recall set in only a fifth of the cases (see Fig. 4.2b). An exception to be observed is that the discovery rate at  $k = 100$  remains 100% no matter the perturbation magnitude. This can be explained by the fact that at these values of  $k$ , the search casts a very wide net which catches at least one photo of the protected. However, as can be seen in Fig. 4.2c, at  $\epsilon \geq 0.06$ , almost all photos in such large recall sets belong to different individuals (identity uniformity is close to 0.0). Therefore, this defense strategy successfully achieves its goal of preserving the privacy of the protected individuals.

**Targeting the Mean of the Lookup Set.** While targeting a randomly sampled lookup set photo is successful, it does come with some downsides, as discussed in section 4.4.3. Therefore, we also

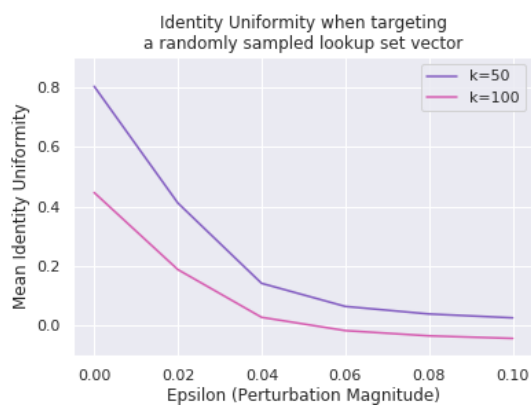




(a) Recall when targeting a randomly sampled lookup set vector



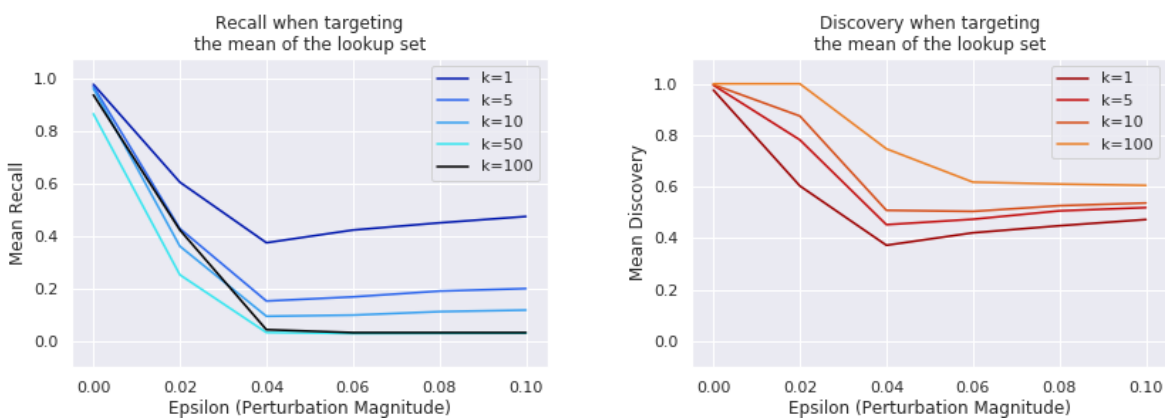
(b) Discovery when targeting a randomly sampled lookup set vector



(c) Identity uniformity when targeting a randomly sampled lookup set vector

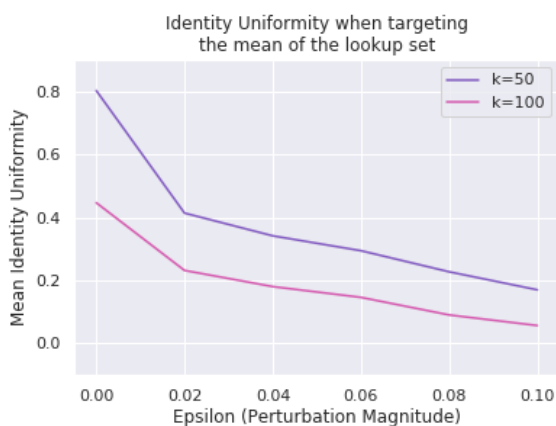
Figure 4.2: Plots of privacy strategy success when targeting a randomly sampled lookup set vector. Observe that perturbation magnitudes of  $\epsilon \geq 0.06$  achieve low recall, low discovery, and high identity uniformity, thereby successfully preserving the privacy of the protected individuals.

experiment with using the mean of the lookup set as a target. Comparing every panel of Fig. 4.3 to every panel of Fig. 4.2 reveals that this targeting strategy is not as effective. For any given combination of  $\epsilon$  and  $k$  values, targeting a randomly selected photo of the lookup set of the protected yields more effective decoys. Recall is between 10 and 20% higher, indicating that there's more



(a) Recall when targeting the mean of the lookup set

(b) Disc. when targeting the lookup set mean.



(c) Identity unif. when targeting the mean of the lookup set

Figure 4.3: Plots of privacy strategy success when targeting the mean of the lookup set. While this defense leaks less information to the protectors and is easier to coordinate, it does not achieve results as good as when targeting a randomly sampled lookup set photo.

photos of the query identity being returned and less decoy photos, on average, in response to queries. Similarly, identity uniformity rises for this same reason.

There is one exception, however. For high values of  $k$  (e.g.,  $k = 100$ ), the discovery rate is consistently lower when targeting the mean of the lookup set (compare Figs. 4.3b and 4.2b). This indicates that targeting the mean does perform one function well — it places decoy photos close to

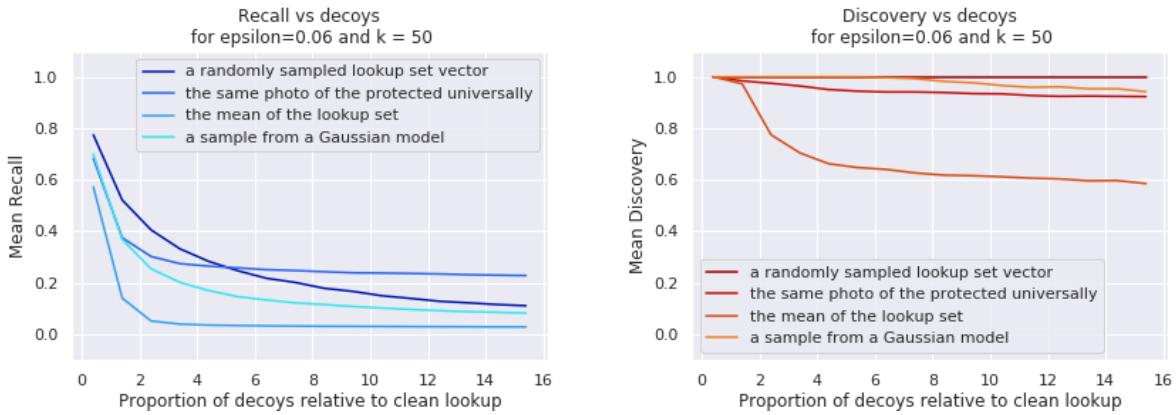
where the query photo lands in the embedding space. Thus, as  $k$  grows, less photos belonging to the protected individual are included in favor of decoy photos. To see this, observe that recall falls with  $k$  in Fig. 4.3a whereas it grows with  $k$  in Fig. 4.2a. Unfortunately, the closest photos to the query do still belong to the protected, thereby hurting the metrics for low values of  $k$  (see the values for  $k = 1, 5, 10$ ).

**Takeaways From All Experiments.** There are several patterns to observe that are common across the experiments with different targeting mechanisms. First, the higher the perturbation magnitude, the more effective the protection scheme is across all metrics and across all targeting approaches. More importantly, the “optimal” value of  $\epsilon$  appears to be 0.06 (see, e.g., Fig. 4.2b; the lowest discovery is achieved at  $\epsilon = 0.06$ ). Increasing the perturbation magnitude to 0.08, or 0.1 only improves the protection scheme by marginal amounts. Thus, to achieve the best tradeoff between degrading image quality and achieving the privacy goals, we recommend using  $\epsilon = 0.06$ .

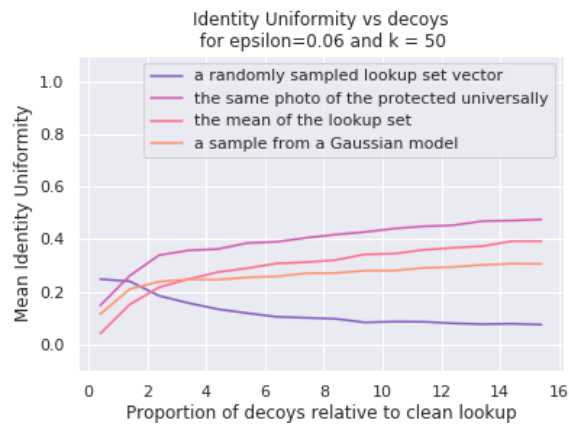
Second, at high  $k$ 's, it is impossible to drive the discovery rate to 0 no matter the perturbation magnitude and the targeting strategy. This is probably because the search casts a very wide net at such high values of  $k$ . However, in terms of privacy, this is not a problem. In fact, at high  $k$ 's, our protection schemes manage to insert a large number of different identities into the top recall set (compare the b and c panels in the figures in this section). When there are many different identities returned in response to a query, the person performing the search through the adversary's services does not know with any reasonable degree of confidence who is depicted in the query. Therefore, the discovery rate is perhaps a bit too harsh and the ultimate goal — of preventing the identification of the person in the query photo — is achieved.

#### 4.6.2 Privacy Protection Success as a Function of the Number of Decoy Photos

We also explored another approach to analyze the effectiveness of the different targeting strategies. The more decoy photos are needed, the harder it is for the privacy protection to succeed. Therefore, we ideally want a targeting strategy that achieves its goal more easily if there are less decoy photos needed. In Fig. 4.4, we present results for  $\epsilon = 0.06$  and  $k = 50$  on this metric. Observe that the



(a) Recall vs. the number of decoy photos as a proportion of  $k$  (b) Discovery vs. the number of decoy photos as a proportion of  $k$



(c) Identity uniformity vs. the number of decoy photos as a proportion of  $k$

Figure 4.4: Graphs of privacy strategy success versus the number of decoy set photos.

recall drops most quickly when targeting the mean of the lookup set. Hence, it might be more desirable to apply this targeting mechanism with a higher  $\epsilon$ . That way, the protection scheme can reap the benefits for discovery rate and identity uniformity discussed in the previous section and achieve them with less decoy photos.

### 4.6.3 Privacy Protection Success When Protectors Do Not Have Access to the Face Recognition Model

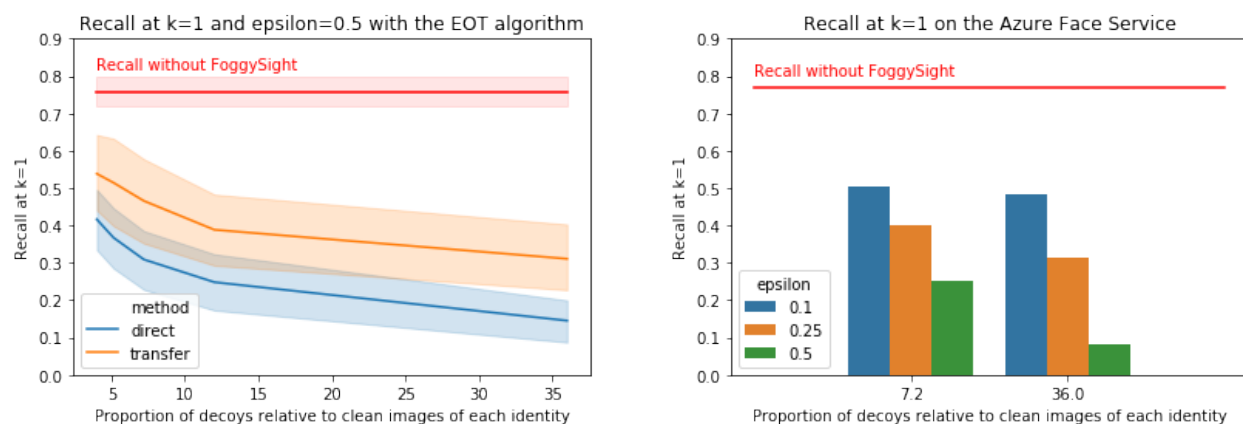
We also explore privacy protections with FoggySight in the scenario where the protectors do not have access to the exact face recognition model used to perform the facial search. As discussed in Section 4.4.2, we adopt two techniques for ensuring that decoys transfer from the model they were generated with to an unknown other model: *Expectation over Transformations* (EOT) for generating robust adversarial examples and ensemble adversarial examples generation. In all experiments in this section, we employ the most successful method from the previous sections – targeting the mean of the Lookup Set.

We first present results on transferability of decoys generated with the EOT algorithm in Figure 4.5a and we give sample decoy images in Figure 4.6. First observe that in both cases, the recall of the network is severely impeded both in the “direct” and the “transfer” cases. The average recall drops below 0.4 with a sufficient number of decoys for both methods. In other words, a protected person has less than a 40% chance of being the nearest neighbor to their query photo – as opposed to 90% chance without the FoggySight defense. This indicates that adversarial example transferability is an effective method to poisoning the facial lookup database to increase individual privacy.

However, we also note that this defense is not 100% effective and that there remains a gap between how effective the “direct” and the “transfer” defenses are. This suggests that stronger methods for generating transferable decoys are needed in order to ensure their effectiveness on unseen models. That is why we explore ensemble generation of adversarial examples and test the results on a commercial face recognition service – one whose internals we do not have access to. In particular, we include both networks implemented in the FaceNet library for our ensemble and measure the results of the scheme on the Azure Face Recognition service.

Results for this transferability to an unseen system are given in Figure 4.5b. They indicate a successful scheme: when  $\epsilon = 0.5$  and there are 36 times more decoys than clean photos, the probability of the service identifying the protected individual is less than 10%. Therefore,

FoggySight can be successful in increasing individual privacy against facial searches, even against unseen systems.



(a) Recall (95% confidence intervals) at  $k = 1$  vs. the number of decoy photos as a proportion of the unaltered photos of an individual.

(b) Recall at  $k = 1$  on the Azure Face Service plotted against the number of decoy photos as a proportion of the unaltered photos of the individual

Figure 4.5: Experimental results when protectors do not have access to the face recognition model

## 4.7 Discussion

**Practical Deployment Considerations** The major step necessary for the effectiveness of FoggySight is wide community adoption. Our experiments – though with a limited set of identities – show that FoggySight requires at least 5 times more decoys than the number of unaltered photos already scraped by the facial search service to reduce the occurrence of the protected identity as a nearest neighbor to the query photo (recall at  $k = 1$ ) to less than 50%. To drive that number even further down to less than 10% on a commercial face service, large perturbation amounts and 36 times more decoys than clean photos are required.

Based on these results, we believe FoggySight is best suited when used to frustrate facial search and create plausible deniability about who a person in a query photo is. With enough decoys, many different identities are returned as a response to a facial search and the true one comprises a small

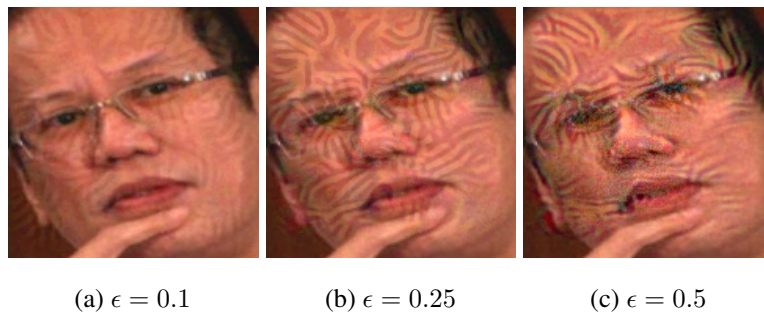


Figure 4.6: Illustration of final transferable decoy images under different perturbation magnitudes  $\epsilon$ . These are images of subject n000957 in the VGGFace2 dataset modified to serve as decoys for other identities.

portion of them. Thus, users of the facial search service cannot be sure with a high degree of confidence that the person in the query photo is any one person from the recall set. This level of protection is reasonable for the individuals similar to those represented in our dataset who may wish to increase their general level of privacy. However, it is absolutely not sufficient for users wishing to prevent discovery. The best solution for those users remains to not have their photos included in the database in the first place or to prevent query photos of themselves from being useful (e.g., through blurring or more advanced obfuscation approaches).

**Facial Search Service Countermeasures (Adaptive Privacy Adversaries)** Our method relies on the ability of adversarial examples to affect the output of the facial search provider’s neural network model and on their ability to remain undetected. There has been research on providing a variety of defenses to adversarial examples. Some of it has shown qualified empirical success [88], some has provided certification guarantees about very specific adversaries [149] and some has even focused on top- $k$  classification [67]. However, the adversarial examples research literature has also found that robust performance (on adversarial examples) often comes at the cost of clean performance (on regular test set examples) [65]. Therefore, we believe it is unlikely that robust neural networks are going to be applied at scale for facial search, as that will trade off the system’s overall reliability on unprotected and protected individuals alike. Furthermore, it is possible that the facial search provider detects and filters some of our decoys. We believe this is out of scope for

our proposal and future work should aim to quantify the effectiveness of such out-of-distribution detection. Our scheme remains effective as long as the ratios of decoys to clean images of a given protected individual can be maintained.

**Incentives and Risks for Protectors** In volunteering to provide decoys, protectors increase their own privacy but also take on an added level of risk. First, even with FoggySight, protected individuals have an incentive to modify any future public photos of themselves so that face recognition models produce embeddings away from their “true” region in the space. This helps maintain the ratio of adversarial to clean images in the facial search provider’s database. The fewer “clean” images the facial search provider has, the harder it is to identify an individual. Thus, protected individuals continue to have an incentive to also serve as protectors for others and participate in FoggySight actively – as opposed to merely receiving protections. We emphasize that this is different from the finding that individuals cannot achieve meaningful protections on their own. In Appendix B.3, we explored cases where individuals wishing privacy modify their future photos in an arbitrary direction and found that that is not enough to increase privacy, given a clean query photo and some clean lookup set photos. FoggySight suggests that they should instead modify their photos in a specific direction.

This, however, introduces a risk for the protector. If a protector participates with an unbalanced number of decoys targeted at a given protected individual, the user of the facial search tool may misidentify the protected as the unbalanced protector. However, this risk can be mitigated by centralized coordination among protectors so that no single one of them is providing a larger-than-average proportion of the decoys for a given protected individual.

**Untagging and Other Defenses against Facial Search** FoggySight is not meant to be a standalone solution. In fact, the less clean photos any given user has in a database, the better decoy-based protection will work for them. Thus, individuals wishing to increase their facial privacy should continue to untag, take down, or otherwise delist their photos from the public Internet. However, we also note that none of these solutions can succeed on its own, either. Reports on facial search providers [57] suggest that millions of individuals already have faces in those databases with links to their (possibly cached) online presence. No amount of untagging, delisting, or removal of photos



can remedy this. FoggySight aims to remedy that through poisoning the database of the facial search provider and is aided by future untagging but neither solution can work on its own.

**Dataset Limitations** While we believe the work in this paper establishes a proof of concept for a collaborative defense approach, all our findings are subject to the limitations of our dataset. For reasons of constrained computational resources, we have worked with a random sample of a bigger dataset of faces that is standard in facial recognition research (see Section 4.5) and our results inherit all limitations of the original dataset. Furthermore, we acknowledge that for full deployment of FoggySight, the scheme would need to undergo rigorous at-scale testing and evaluation. In particular, such testing needs to ensure that different populations of users are represented properly and that protections apply to every group equally well – and especially to groups that may suffer worse consequence of diminished facial privacy than others. We further refer the reader to recent works on ethical auditing of face recognition technology [16] and encourage future work that also considers facial search protection works, such as this one.

**Impact of Transferable Adversarial Examples (Decoys)** In our experiments, we found that FoggySight protectors need to introduce both higher-magnitude perturbations to their images and provide more decoys when they do not have access to the adversary’s model. For example, where protectors acting with access to the facial search model needed to inject 2-4 times more decoys with  $\epsilon = 0.04$  than unaltered, previously scraped images of the protected, protectors need to inject 36 times more decoys with  $\epsilon = 0.5$  to be really effective against commercial face recognition services with unknown internals. This suggests that a potential policy response that may enable individuals to apply FoggySight more effectively might be to mandate disclosure of the facial search model. The *best* policy responses to facial search adversaries are beyond the scope of this work, but we highlight this finding as a possible remediation mechanism that may provide individuals with more agency in protecting their privacy.

## Chapter 5

# DISRUPTING MODEL TRAINING WITH ADVERSARIAL SHORTCUTS

### *5.1 Introduction*

Machine learning capabilities have not always been put to good use. The ubiquity of face recognition technology has reduced the privacy of individuals using web services to share photos. Generative models have blurred the line between real and fake content. And the failures of machine learning have disproportionately affected historically marginalized communities. Even when the use of machine learning does not cause harm to society and privacy, not all stakeholders in the creation of a functioning model agree to participate in that process.

This tension is most obvious in the compilation and use of image datasets. Visual data shared online often gets applied for machine learning uses that the data creator did not envision. For example, copyright holders do not always consent to having their content included in the training set of models. Additionally, individuals who share facial photos online do not wish for those photos to be used to train biometric models or – worse – for “deepfake” images of them to be created and used to frame them for embarrassing or illegal activity. However, in many of these scenarios, publishers of content do wish for that content to be useful to other humans. Social media users share their photos for the enjoyment of their friends and family. Professional photographers post pictures online in order to promote their work and make it available for purchase.

In this chapter, we explore how creators and/or publishers of visual content can prevent unauthorized uses of their data for machine learning while retaining the data’s utility to other humans. One approach to achieving this goal is to prevent the automated collection of such data by unauthorized parties. For example, social media companies and photo sharing websites deploy technical mechanisms against scraping and they disallow such activity in their legal agreements with users. However, both of those methods only partially achieve the goal. Despite anti-scraping technology,

unauthorized third parties have been able to collect large datasets of photos from social media: for instance, Clearview.AI collected millions of facial images of individuals from a variety of online services [57]. Legal agreements, in turn, cannot be enforced on a technical level once the data has been scraped and rely on legal compliance, which cannot always be guaranteed (e.g., if the party training the model is in a different jurisdiction than the data creator).

We, therefore, wish to develop methods for protecting datasets from being used to train machine learning models for undesired purposes even when untrusted parties can obtain the data. Because this is a broad and challenging problem, our initial focus is the canonical setting of multi-class image classification. Our goal is to modify a clean dataset so that ML models, and primarily deep neural networks (DNNs), achieve high training accuracy while failing to generalize to unmodified test examples. As a mechanism for such modifications, we explore *adversarial shortcuts*, a method that encourages DNNs to lazily rely on spurious signals rather than robust, semantic features.

The idea of adversarial shortcuts is partially inspired by earlier work on blindspots in CNN model training. First, studies have observed that neural networks often leverage non-semantic features to achieve good test set performance. For example, high-frequency features and non-robust features correlated with the true class can help explain the ability of neural networks to generalize to their test set, even though such features do not correspond to human understanding [88]. Second, prior work has shown that convolutional classifiers can use spurious correlations that are “simpler” than the true semantics in order to output predictions. [26, 122] This is most evident in failures of the models: for example, animals on green backgrounds are more often classified as cows (even if they are something else) with models that were trained on datasets with only cows on green backgrounds. The green background appears to be a stronger and simpler signal for the model than the detailed features of the particular animal. In this chapter, we identify a potential use case of these observations as a security measure that prevents training.

## 5.2 Setup and Goals

In order to aid further discussion, we begin with a few formal definitions of our setup and goals. Assume that we are in possession of a dataset  $D_{train} = \{(x_i, y_i)\}_{i=1}^n$  where  $x_i \in \mathbb{R}^{w \times h \times c}$  are RGB

images of width  $w$ , height  $h$  and channels  $c$  and  $y_i \in [0, \dots, N]$  are their corresponding true, semantic labels (expressed as indices in a list of  $N$  classes) for a classification task.  $D$  is assumed to be drawn from a true data-generating distribution  $\mathcal{D}$ . As in standard supervised learning, we assume that models trained on  $D_{train}$  will be used to classify samples  $D_{test}$  drawn independently from  $\mathcal{D}$ . We denote such models with parameters  $\theta$  as  $f_\theta$  and assume that  $\theta$  is obtained via the standard training procedure using stochastic gradient descent to achieve

$$\theta^* = \arg \min_{\theta} \sum_i L(f_\theta(x_i), y_i)$$

for a loss function  $L$  measuring the distance between the outputs of  $f$  and the true labels (such as cross-entropy). Models are further evaluated in terms of their accuracy on the test data:

$$Acc(f, \theta, D_{test}) = \frac{1}{|D_{test}|} \sum_{D_{test}} \mathbb{1}(\arg \min f_\theta(x_i) = y_i)$$

We wish to create a protected dataset  $D' = \{(x'_i, y_i)\}$  such that:

- **Semantics in  $D'$  are preserved.**

The modified images  $x'_i$  differ from their corresponding original images  $x_i$  minimally. In other words,  $dist(x_i, x'_i)$  is low for a given distance metric  $dist$ . The labels  $y_i$  are not modified from their original version in  $D_{train}$  and correspond to the true semantics of  $x_i$ . We assume that a party obtaining our modified dataset may reconstruct  $y_i$  even if we do not provide them.

- **Models trained on  $D'$  achieve low test accuracy.**

If we obtain  $\theta'$  such that  $\theta' = \arg \min_{\theta'} \sum_i L(f_{\theta'}(x_i), y_i)$  is achieved,  $f_{\theta'}$  only achieves low test accuracy:  $Acc(f, \theta', D_{test}) \ll Acc(f, \theta, D_{test})$  for  $\theta$  obtained from the unmodified  $D_{train}$ .

### 5.3 Proposed Methods for Protective Dataset Modifications

In this section, we introduce dataset modifications that encourage CNNs to rely on spurious signals rather than robust, semantic features. We propose three approaches: a sparse pixel-based pattern, a

visible watermark, and a brightness modulation. All three generate modifications that are unique to each class  $k \in \{1, \dots, K\}$ , creating a shortcut that the DNN can use to quickly achieve high accuracy on the training data while failing to generalize to unmodified examples. We refer to such modifications as *adversarial shortcuts*, and each technique is tuneable, allowing us to control the tradeoff between disrupting training and preserving visual features in the data.

### 5.3.1 Sparse Pixel-Based Patterns

The first approach we introduce is a sparse, pixel-based modification. We generate random perturbation masks  $\Delta_k \in \{0, 1\}^{w \times h \times c}$  for each class  $k \in \{1, \dots, K\}$  with entries determined as follows: for each value, we sample  $\delta \sim \mathcal{N}(\mu, \sigma)$  and set the value to one if  $\delta$  exceeds the middle of the pixel brightness range (e.g., 0.5 with 0-1 normalization), and zero otherwise. In practice, we fix  $\sigma = 0.2$  and experiment with different  $\mu$  values.

With the masks fixed, we then modify images in the pixels indicated by their corresponding perturbation masks. Assuming that the maximum pixel value in the dataset is given by  $x_{\max} \in \mathbb{R}$ , we generate the modified image  $x'_i$  with label  $y_i = k$  using the formula

$$x'_i = (1 - \Delta_k) \odot x_i + \Delta_k \cdot x_{\max}. \quad (5.1)$$

Examples from the CIFAR10 dataset are given in Figure C.3 and an ImageNet-sized image with these modifications at  $\mu = 0.01$  is given in Figure 5.1b.

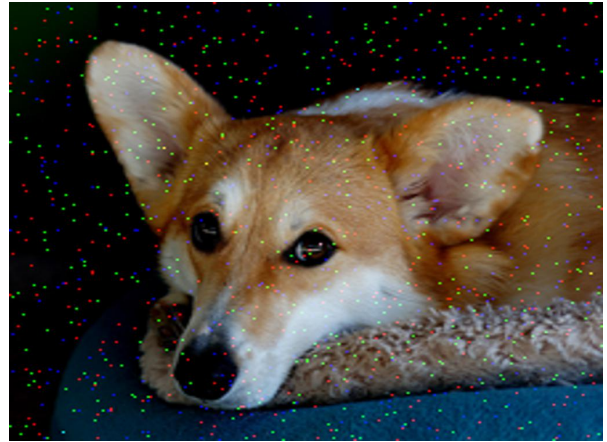
### 5.3.2 Visible Watermarks

The next approach we introduce is a visible, class-specific watermark. If the watermark is prominent and easy to detect, a DNN can use it as a shortcut to achieve high accuracy without relying on robust features. To efficiently generate shapes with a sufficient degree of variation, which is known to make watermarks more difficult to remove [27], we create watermarks by enumerating the class indices using digits from the MNIST dataset [79].

For example, in CIFAR-10 [76] the “airplane” class has index 0, so we create a watermark for



(a) Original, unmodified  
image



(b) Image modified with pixel-based approach with  $\mu = 0.01$



(c) Image modified with visual watermark approach with  $\alpha = 0.50$



(d) Image modified with brightness modulation approach with  $\gamma = 0.90$

Figure 5.1: Example of an ImageNet-sized image with the various modification techniques applied. The image depicted here was originally available at <https://www.flickr.com/photos/volvob12b/9797687423>, was accessed on June 3, 2021, and is distributed in the Public Domain. To the best of our knowledge, this image is not actually part of the ImageNet dataset but if it were, it would have class index 263 for ‘Pembroke, Pembroke Welsh corgi.’

each airplane example by randomly sampling a zero from the MNIST dataset. For ImageNet, which has 1000 classes, the watermarks require up to three randomly selected digits. The watermark generation process for each class  $k \in \{1, \dots, K\}$  can be understood as sampling a binary image  $M \in \{0, 1\}^{w \times h \times c}$  from a random variable  $\mathcal{M}(k)$ , which we then blend with the original image  $x_i$  using a parameter  $\alpha \in [0, 1]$  as follows:

$$x'_i = \alpha \cdot M + (1 - \alpha) \cdot M \cdot x_i + (1 - M) \cdot x_i. \quad (5.2)$$

The blending parameter  $\alpha$  controls how visible the watermark is, with  $\alpha = 0$  having no effect and  $\alpha = 1$  overlaying the watermark on the original image. An example with  $\alpha = 0.5$  is shown in Figure 5.1c, with index 263 for the “Pembroke, Pembroke Welsh corgi” class.

### 5.3.3 Brightness Modulation Patterns

While the previous two approaches provide shortcuts that can successfully disrupt model training, they may prove easy to remove with basic countermeasures. Our next approach is designed to be more difficult to circumvent. Rather than creating a localized, visually distinguishable perturbation, we now modify images using a randomized brightness modulation that either brightens or darkens pixels identically for images in each class.

The brightness modulation for each class is generated as follows. At the start, we randomly sample a location in the image that serves as the center of a square; we then decide, with equal probability, whether to darken or brighten the corresponding pixels. Given a parameter  $\gamma \in [0.5, 1]$ , we darken pixels by multiplying them by  $\gamma$  or brighten them by multiplying by  $2 - \gamma$ . We perform  $T$  iterations of this process with  $T$  distinct squares, which can and do overlap, resulting in a checkerboard-type pattern.

This process is equivalent to sampling a class-specific mask  $B_k \in \mathbb{R}^{w \times h \times c}$ , where an image  $x_i$  with class  $y_i = k$  is modified using the following formula:

$$x'_i = B_k \odot x_i. \quad (5.3)$$

An example of this modification is shown in Figure C.5 with the parameter  $\gamma = 0.7$  and  $T = 600$  iterations. For all experiments with ImageNet we set  $T = 600$ , and for CIFAR-10 we use  $T = 32$ .

## 5.4 Experimental Setup

We experiment with two standard computer vision datasets: CIFAR10 [76] and ImageNet [113]. In both cases, we apply our protective modifications to the training set and keep the validation set intact. By measuring accuracy on the validation set, we can observe how well the trained models solve their intended classification task. If our protections are successful, the best achievable accuracy would be low.

For the experiments with CIFAR10, we run training for 50 epochs at a batch size of 1024 and vary the learning rate, the architecture of the classifier used, and the random seed. Specifically, we experiment with learning rates 0.1, 0.01, 0.001, 0.0001; with classifiers ResNet18 [53], DenseNet201 [60], VGG11 [126], and SqueezeNet [63]; and with seeds 3525462, 15254521, 63246662, 32542462. We then report the best achievable accuracy across all the runs for several different settings of each of our intensity parameters ( $\mu$  for pixel-based patterns,  $\alpha$  for watermarking-based patterns, and  $\gamma$  for the brightness modulation pattern). All networks are as implemented in torchvision [108] and are trained from scratch, with no pretraining.

For ImageNet, we use the training script available at <https://github.com/pytorch/examples/tree/master/imagenet> and the ResNet18 [53] architecture. We only train at default hyperparameter values due to our limited computational resources.

## 5.5 Evaluation

In this section, we evaluate our proposed techniques for disrupting model training. Although we do not reach state-of-the-art training accuracy on CIFAR-10 and ImageNet, either due to computational constraints or insufficient hyperparameter tuning, we ensure a fair comparison by using identical training procedures across all experiments.



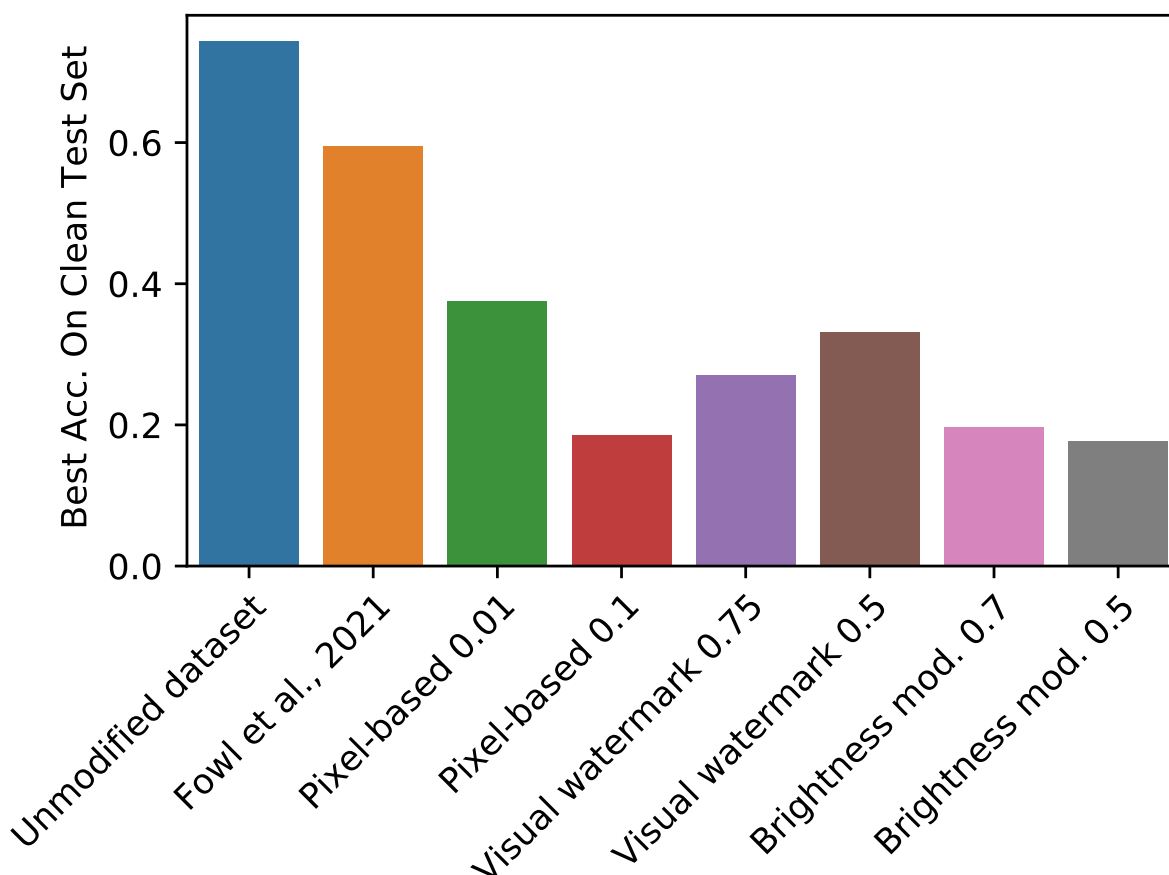


Figure 5.2: Best achievable test accuracy after 50 epochs when training ResNet18 on CIFAR-10 with different dataset modifications.

### 5.5.1 Training on modified CIFAR-10

We first test our dataset modifications on CIFAR-10. Figure 5.2 summarizes the results from training a ResNet18 architecture with various dataset modifications. Figures C.1a, C.1b, and C.1c provide more ablations and details, including different model architectures and more parameter settings for each adversarial shortcut. The best achievable validation accuracy after 50 epochs with the unmodified version of CIFAR-10 is above 70% accuracy, while all of our modifications, even at the weakest settings we tested, have a significant impact on model performance.

Relatively small pixel-based perturbations are enough to nearly halve the accuracy: with  $\mu = 0.01$ , CIFAR-10 classifiers achieve at best no more than 40% accuracy. This setting corresponds to modifying only 22 out of 3,072 pixels, on average. Similarly, visible watermark perturbations with a blend factor of  $\alpha = 0.5$  and brightness modulations with parameter  $\gamma = 0.9$  succeed in reducing the validation accuracy to less than 40%.

Fowl et al. [37] shared a version of CIFAR-10 protected with their proposed approach, which we compared with our methods.<sup>1</sup> While the validation accuracy does not reach 70%, as with training on the unprotected version of the dataset, models trained on the dataset with [37] protections manage to achieve up to 60% accuracy, which is significantly higher than our methods (also see Figure C.1d).

We also tested the stability of our proposed modifications for disrupting training when certain countermeasures are in place. For this purpose, we considered two categories of countermeasures: aggressive training set augmentations and the addition of Gaussian noise. Results for the pixel-based approach with these countermeasures are shown in Figures C.2b, C.2c and C.2d. Our findings generalize, and the best achievable accuracy across the same set of hyperparameters and random seeds remains the same. However, for the brightness modulation method with  $\gamma = 0.9$ , aggressive augmentations are effective at undoing the modifications and allowing effective training (see Figure C.2d). We suggest using a stronger setting of  $\gamma = 0.70$  that makes the brightness modulation more visible.

### 5.5.2 Training on modified ImageNet

Next, we perform experiments on ImageNet. Although we do not have the computational resources to train with a variety of hyperparameter and random seed choices, several takeaways are apparent from Figure 5.3a and additional results in Figure 5.3b. First, sparse pixel-based pattern protections and visible watermark protections remain effective. In both cases, the best achievable validation accuracy is less than 30%, whereas training on the unprotected version of ImageNet easily achieves more than 50% accuracy with the same setup. This is again achievable with fairly

---

<sup>1</sup>The version of their defense that the authors shared with us for this test has parameters  $\epsilon = 8/255$ , and we note that stronger training disruptions may be achieved with different parameters.

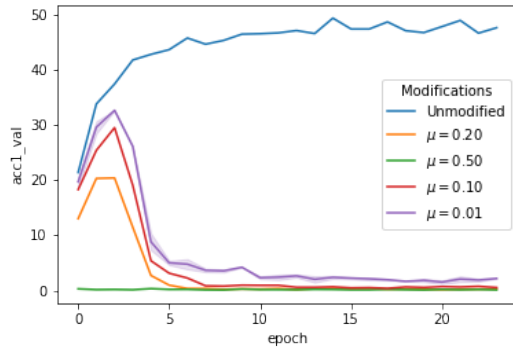
minor modifications, such as pixel-based with  $\mu = 0.01$  and visible watermarking with  $\alpha = 0.5$ .

Furthermore, the plots of accuracy on the clean validation set and the protected training set in Figure 5.3c reveal an interesting dynamic. While the model can fit the training set extremely well, achieving up to 90% accuracy, it does not generalize to the validation set. This suggests that our objective of disrupting training with a non-robust shortcut is successful, and that the models fits the simple class-specific pattern as opposed to the true semantics. This divergence in training and validation accuracy, or the rapid increase in the generalization gap, does not manifest when training with the unmodified ImageNet data (Figure 5.3c).

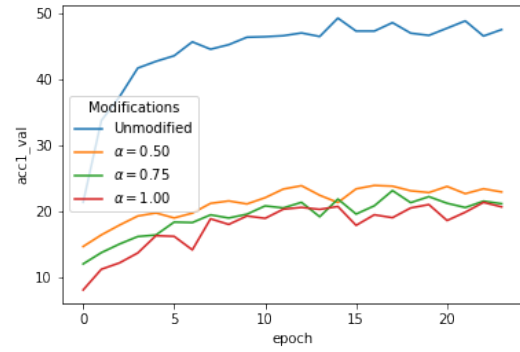
## **5.6 Discussion**

Our experiments show that it is possible to disrupt DNN training by modifying datasets with simple patterns, such as our adversarial shortcuts, that discourage models from relying on robust, generalizable features. These modifications can reduce model accuracy on clean data while having minimal impact on the image semantics. Our work focuses on the narrow setting of multi-class image classification, but there is great potential for future work that considers more effective dataset modifications, attempts to undo protective modifications, and develops new approaches for different ML tasks, e.g., preventing the unauthorized development of deepfakes or facial recognition systems.

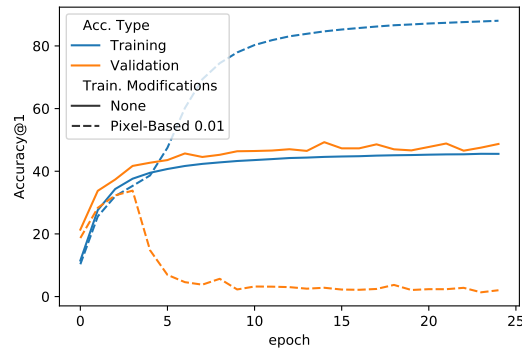
We hope that future work considers how to remove adversarial shortcuts, develops tools to compare different methods' tradeoffs between preserving semantics and disrupting training, and that these insights help inform more resilient methods. We are confident that, over time, such attack/defend iterations will lead to robust measures to disrupt training on a variety of tasks.



(a) Validation accuracy at different levels of pixel-based protections



(b) Validation accuracy at different levels of visible watermarking protections



(c) ResNet18 train and validation accuracy on ImageNet protected with the pixel-based modification at  $\mu = 0.01$ , compared with unmodified ImageNet. Validation accuracy can reach up to 70% with unmodified data, but our modified dataset prevents effective learning within the first several epochs.

Figure 5.3: Validation accuracy progress when training ResNet18 on a protected ImageNet with standard augmentations during training.

## Chapter 6

### CONCLUSION

In Chapter 3, we introduced an algorithm ( $RP_2$ ) that generates robust, physically realizable adversarial perturbations. Using  $RP_2$ , and a two-stage experimental design consisting of lab and drive-by tests, we contribute to understanding the space of physical adversarial examples when the *objects themselves* are physically perturbed. We target road-sign classification because of its importance in safety, and the naturally noisy environment of road signs. Our work shows that it is possible to generate physical adversarial examples robust to widely varying distances/angles. This implies that future defenses should not rely on physical sources of noise as protection against physical adversarial examples.

Furthermore, companies today are scraping photos from social media sites and are using those photos to build powerful systems capable of identifying people from newly taken photos [55]. Therefore, in Chapter 4, we proposed FoggySight, a community-based approach for modifying future photos provided publicly on the Internet so that they crowd out previously scraped photos. Our experiments demonstrate that FoggySight can meaningfully increase privacy. As with any early proposal, many practical questions need to be answered for full deployment and desired effectiveness. However, we are convinced that this work both highlights the limitations of facial privacy protection schemes and proposes a solid basis for future work in this space to build on.

Finally, in Chapter 5, we posit that there are settings where it is desired that machine learning training not succeed, such as when data owners want to prevent unauthorized uses of their data. For those situations, we develop and study a set of modifications to training sets that prevent state-of-the-art models from achieving meaningful classification accuracy on the true distribution. Our results suggest that hand-crafted approaches might be better for achieving the goal of dataset protection than gradient-based approaches, such as those applied for data poisoning. Taken together,

these three studies address important issues of security and privacy when machine learning models are used for computer vision.

As the technology continues to evolve and new uses for it are discovered, these issues will likely evolve as well. On the security side of the duality, machine learning models applied for computer vision are likely to remain vulnerable to a range of attacks. In addition to digital and physical adversarial perturbations, more accessible attack vectors remain a threat to current computer vision systems [43]. Additionally, novel architectures and self-supervised learning are emerging as potential next steps in the growth of the field [21] but they have also proven to be susceptible to fairly counterintuitive attacks. For example, recent work has shown that deep neural networks can learn to associate visual and textual concepts [44]. This promises to align neural network processing closer with human reasoning. Yet such multimodally trained models fall pray to so-called “typographic attacks,” where the model prefers to use a pen-and-paper label in the image for its prediction over the shape and texture of the object at hand. For these reasons, researchers and practitioners should approach the security of machine learning as a systems level problem. They should consider how appropriate safeguards can be built around vulnerable models for their concrete application scenario and think of securing the system as a whole rather than relying on the (possibly unreachable) infallibility of the model itself. I am convinced that a continuation of the attack/defense cycle of finding novel vulnerabilities and not-so-sophisticated blindspots in models and designing systems to be secure in the face of such failures is the best path forward to improving the security of computer vision systems.<sup>1</sup>

Unfortunately, technology is never solely a force for good and its “proper” operation can cause harm in some cases and machine learning is no exception. However, the imperfection in technology also offers the ability to reduce the negative effects of new technological capabilities. I am excited to see how the ideas studied in Chapters 4 and 5 can be expanded to provide stronger guarantees for privacy and protection against unwanted uses of data for machine learning. Moreover, I believe that technological advancements in this space will ultimately need to be backed by cultural and legal

---

<sup>1</sup>For further discussion of this topic, see [35]

norms. If we consider facial search technology to be too invasive and dangerous, then we should regulate and perhaps restrict its use, in addition to disrupting its operation on a technical level. If we want to enforce what our data can be applied for, we need to have the legal basis for asserting our ownership, in addition to preventing the training of machine learning models on it.

As machine learning gets applied in more and more different fields, new security and privacy issues may emerge. I look forward to doing and reading the research that explores those in the future.

## BIBLIOGRAPHY

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from [tensorflow.org](https://www.tensorflow.org).
- [2] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.
- [3] Hojjat Aghakhani, Dongyu Meng, Yu-Xiang Wang, Christopher Kruegel, and Giovanni Vigna. Bullseye polytope: A scalable clean-label poisoning attack with improved transferability. *arXiv preprint arXiv:2005.00191*, 2020.
- [4] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 28(12):2037–2041, 2006.
- [5] Kendra Albert, Jon Penney, Bruce Schneier, and Ram Shankar Siva Kumar. Politics of adversarial machine learning. In *Towards Trustworthy ML: Rethinking Security and Privacy for ML Workshop, Eighth International Conference on Learning Representations (ICLR)*, 2020.
- [6] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.
- [7] Anish Athalye. Robust adversarial examples. <https://blog.openai.com/robust-adversarial-inputs/>, 2017.
- [8] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.



- [9] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*, 2017.
- [10] Samyadeep Basu, Philip Pope, and Soheil Feizi. Influence functions in deep learning are fragile. *arXiv preprint arXiv:2006.14651*, 2020.
- [11] Manuele Bicego, Andrea Lagorio, Enrico Grosso, and Massimo Tistarelli. On the use of sift features for face authentication. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 35–35. IEEE, 2006.
- [12] Battista Biggio, Blaine Nelson, and Pavel Laskov. Support vector machines under adversarial label noise. In *Asian conference on machine learning*, pages 97–112, 2011.
- [13] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on International Conference on Machine Learning, ICML'12*, pages 1467–1474, USA, 2012. Omnipress.
- [14] Haitham Bou-Ammar, Holger Voos, and Wolfgang Ertel. Controller design for quadrotor uavs using reinforcement learning. In *Control Applications (CCA), 2010 IEEE International Conference on*, pages 2130–2135. IEEE, 2010.
- [15] Vicki Bruce and Andy Young. Understanding face recognition. *British journal of psychology*, 77(3):305–327, 1986.
- [16] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.
- [17] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 67–74. IEEE, 2018.
- [18] Nicholas Carlini, Samuel Deng, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmood, Shuang Song, Abhradeep Thakurta, and Florian Tramèr. An attack on instahide: Is private learning possible with instance encoding? *arXiv preprint arXiv:2011.05315*, 2020.
- [19] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14, 2017.
- [20] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

- [21] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.
- [22] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.
- [23] Valeriia Cherepanova, Micah Goldblum, Harrison Foley, Shiyuan Duan, John P Dickerson, Gavin Taylor, and Tom Goldstein. Lowkey: Leveraging adversarial attacks to protect social media users from facial recognition. *OpenReview*, 2020.
- [24] Moustapha Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. Houdini: Fooling deep structured prediction models. *arXiv preprint arXiv:1707.05373*, 2017.
- [25] Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108, 2004.
- [26] Alex J DeGrave, Joseph D Janizek, and Su-In Lee. Ai for radiographic covid-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, pages 1–10, 2021.
- [27] Tali Dekel, Michael Rubinstein, Ce Liu, and William T Freeman. On the effectiveness of visible watermarks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2146–2154, 2017.
- [28] Pieter Delobelle, Paul Temple, Gilles Perrouin, Benoît Frénay, Patrick Heymans, and Bettina Berendt. Ethical adversaries: Towards mitigating unfairness with adversarial machine learning. *arXiv preprint arXiv:2005.06852*, 2020.
- [29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
- [30] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.
- [31] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7714–7722, 2019.

- [32] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.
- [33] EFF. Law enforcement use of face recognition systems threatens civil liberties, disproportionately affects people of color: Eff report. February 2018.
- [34] Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft. Privacy-preserving face recognition. In *International symposium on privacy enhancing technologies symposium*, pages 235–253. Springer, 2009.
- [35] Ivan Evtimov, Weidong Cui, Ece Kamar, Emre Kiciman, Tadayoshi Kohno, and Jerry Li. Security and machine learning in the real world. *arXiv preprint arXiv:2007.07205*, 2020.
- [36] Minghong Fang, Neil Zhenqiang Gong, and Jia Liu. Influence function based data poisoning attacks to top-n recommender systems. In *Proceedings of The Web Conference 2020*, pages 3019–3025, 2020.
- [37] Liam Fowl, Ping-yeh Chiang, Micah Goldblum, Jonas Geiping, Arpit Bansal, Wojtek Czaja, and Tom Goldstein. Preventing unauthorized use of proprietary data: Poisoning for secure dataset release. *arXiv preprint arXiv:2103.02683*, 2021.
- [38] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.
- [39] Chuhan Gao, Varun Chandrasekaran, Kassem Fawaz, and Somesh Jha. Face-off: Adversarial face obfuscation. *arXiv preprint arXiv:2003.08861*, 2020.
- [40] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.
- [41] Jonas Geiping, Liam Fowl, W Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. Witches’ brew: Industrial scale data poisoning via gradient matching. *arXiv preprint arXiv:2009.02276*, 2020.
- [42] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

- [43] Justin Gilmer, Ryan P Adams, Ian Goodfellow, David Andersen, and George E Dahl. Motivating the rules of the game for adversarial example research. *arXiv preprint arXiv:1807.06732*, 2018.
- [44] Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 6(3):e30, 2021.
- [45] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Data security for machine learning: Data poisoning, backdoor attacks, and defenses. *arXiv preprint arXiv:2012.10544*, 2020.
- [46] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- [47] Thore Graepel, Kristin Lauter, and Michael Naehrig. Ml confidential: Machine learning on encrypted data. In *International Conference on Information Security and Cryptology*, pages 1–21. Springer, 2012.
- [48] Patrick Grother, Mei Ngan, and Kayee Hanaoka. Ongoing face recognition vendor test (frvt) part 2: Identification. *National Institute of Standards and Technology, Tech. Rep*, 2018.
- [49] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.
- [50] Ruocheng Guo, Pengchuan Zhang, Hao Liu, and Emre Kiciman. Out-of-distribution prediction with invariant risk minimization: The limitation and an effective fix. *arXiv preprint arXiv:2101.07732*, 2021.
- [51] Drew Harwell. This facial recognition website can turn anyone into a cop — or a stalker. *The Washington Post*, May 2021. Accessed: 2021-05-24.
- [52] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [54] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang. Face recognition using laplacianfaces. *IEEE transactions on pattern analysis and machine intelligence*, 27(3):328–340, 2005.

- [55] Rebecca Heilweil. The world’s scariest facial recognition company explained. *Vox*, May 2020.
- [56] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.
- [57] Kashmir Hill, Jennifer Valentino-DeVries, Gabriel J.X. Dance, and Aaron Krolik. The secretive company that might end privacy as we know it. *New York Times*, Jan 2020.
- [58] Daniel C Howe and Helen Nissenbaum. Engineering privacy and protest: A case study of adnauseam. In *IWPE@ SP*, pages 57–64, 2017.
- [59] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. Discriminative deep metric learning for face verification in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1875–1882, 2014.
- [60] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.
- [61] W Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. Metapoison: Practical general-purpose clean-label data poisoning. *arXiv preprint arXiv:2004.00225*, 2020.
- [62] Yangsibo Huang, Zhao Song, Kai Li, and Sanjeev Arora. Instahide: Instance-hiding schemes for private distributed learning. In *International Conference on Machine Learning*, pages 4507–4518. PMLR, 2020.
- [63] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.
- [64] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. *arXiv preprint arXiv:1804.08598*, 2018.
- [65] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136, 2019.
- [66] Matthew Jagielski, Giorgio Severi, Niklas Pousette Harger, and Alina Oprea. Subpopulation data poisoning attacks. *arXiv preprint arXiv:2006.14026*, 2020.

- [67] Jinyuan Jia, Xiaoyu Cao, Binghui Wang, and Neil Zhenqiang Gong. Certified robustness for top-k predictions against adversarial perturbations via randomized smoothing. *arXiv preprint arXiv:1912.09899*, 2019.
- [68] Jinyuan Jia and Neil Zhenqiang Gong. Attriguard: A practical defense against attribute inference attacks via adversarial machine learning. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 513–529, 2018.
- [69] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 259–274, 2019.
- [70] Kang-Xing Jin. Keeping our platform safe with remote and reduced content review. <https://about.fb.com/news/2020/10/coronavirus/>, March 2020. Online; accessed 29 October 2020.
- [71] Jason Jo and Yoshua Bengio. Measuring the tendency of cnns to learn surface statistical regularities. *arXiv preprint arXiv:1711.11561*, 2017.
- [72] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4873–4882, 2016.
- [73] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894, 2017.
- [74] Stepan Komkov and Aleksandr Petiushko. Advhat: Real-world adversarial attack on arcface face id system. *arXiv preprint arXiv:1908.08705*, 2019.
- [75] Jakub Konečný, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.
- [76] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).
- [77] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

- [78] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.
- [79] Yann LeCun. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998.
- [80] Tao Li and Lei Lin. Anonymousnet: Natural face de-identification with measurable privacy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.
- [81] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- [82] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.
- [83] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*, 2017.
- [84] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. 2017.
- [85] Daniel Lowd and Christopher Meek. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 641–647, 2005.
- [86] Daniel Lowd and Christopher Meek. Good word attacks on statistical spam filters. In *CEAS*, volume 2005, 2005.
- [87] Jiajun Lu, Hussein Sibai, Evan Fabry, and David Forsyth. No need to worry about adversarial examples in object detection in autonomous vehicles. *arXiv preprint arXiv:1707.03501*, 2017.
- [88] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.
- [89] Iacopo Masi, Yue Wu, Tal Hassner, and Prem Natarajan. Deep face recognition: A survey. In *2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, pages 471–478. IEEE, 2018.

- [90] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.
- [91] Ezgi Mercan, Sachin Mehta, Jamen Bartlett, Linda G Shapiro, Donald L Weaver, and Joann G Elmore. Assessment of machine learning of breast pathology structures for automated differentiation of breast cancer and high-risk proliferative lesions. *JAMA network open*, 2(8):e198777–e198777, 2019.
- [92] Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer. Universal adversarial perturbations against semantic image segmentation. *arXiv preprint arXiv:1704.05712*, 2017.
- [93] Andreas Mogelmoose, Mohan Manubhai Trivedi, and Thomas B. Moeslund. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *Trans. Intell. Transport. Sys.*, 13(4):1484–1497, December 2012.
- [94] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. *CoRR*, abs/1610.08401, 2016.
- [95] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. *arXiv preprint arXiv:1511.04599*, 2015.
- [96] Christian Mostegel, Markus Rumpler, Friedrich Fraundorfer, and Horst Bischof. Uav-based autonomous image acquisition with multi-view stereo quality assurance by confidence prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–10, 2016.
- [97] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wonggrasamee, Emil C Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 27–38. ACM, 2017.
- [98] Luis Muñoz-González, Bjarne Pfitzner, Matteo Russo, Javier Carnerero-Cano, and Emil C Lupu. Poisoning attacks with generative adversarial nets. *arXiv preprint arXiv:1906.07773*, 2019.
- [99] Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. *arXiv preprint arXiv:2010.15775*, 2020.
- [100] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015.



- [101] Nicholas Nuechterlein and Sachin Mehta. 3d-espnet with pyramidal refinement for volumetric brain tumor image segmentation. In *International MICCAI Brainlesion Workshop*, pages 245–253. Springer, 2018.
- [102] Seong Joon Oh, Mario Fritz, and Bernt Schiele. Adversarial image perturbation for privacy protection a game theory perspective. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1491–1500. IEEE, 2017.
- [103] Tess Owen. White supremacists built a website to doxx interracial couples — and it’s going to be hard to take down. *Vice*, May 2020.
- [104] Nicolas Papernot, Ian Goodfellow, Ryan Sheatsley, Reuben Feinman, and Patrick McDaniel. cleverhans v1.0.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 2016.
- [105] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.
- [106] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.
- [107] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. *arXiv*, 2015.
- [108] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [109] Christine I Podilchuk and Edward J Delp. Digital watermarking: algorithms and applications. *IEEE signal processing Magazine*, 18(4):33–46, 2001.
- [110] Arezoo Rajabi, Rakesh B Bobba, Mike Rosulek, Charles V Wright, and Wu-chi Feng. On the (im) practicality of adversarial perturbation for image privacy. *Proceedings on Privacy Enhancing Technologies*, 2021(1):85–106.

- [111] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [112] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.
- [113] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- [114] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Radioactive data: tracing through training. In *International Conference on Machine Learning*, pages 8326–8335. PMLR, 2020.
- [115] Ahmad-Reza Sadeghi, Thomas Schneider, and Immo Wehrenberg. Efficient privacy-preserving face recognition. In *International Conference on Information Security and Cryptology*, pages 229–244. Springer, 2009.
- [116] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. *arXiv preprint arXiv:1910.00033*, 2019.
- [117] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.
- [118] Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? A unified benchmark for backdoor and data poisoning attacks. *arXiv preprint arXiv:2006.12557*, 2020.
- [119] Pierre Sermanet and Yann LeCun. Traffic sign recognition with multi-scale convolutional networks. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2809–2813. IEEE, 2011.
- [120] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suci, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! Targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems*, pages 6103–6113, 2018.
- [121] Shawn Shan, Emily Wenger, Jiayun Zhang, Huiying Li, Haitao Zheng, and Ben Y Zhao. Fawkes: protecting privacy against unauthorized deep learning models. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 1589–1604, 2020.

- [122] Janelle Shane. Do neural nets dream of electric sheep?, Mar 2018.
- [123] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016.
- [124] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. A general framework for adversarial examples with objectives. *ACM Transactions on Privacy and Security (TOPS)*, 22(3):1–30, 2019.
- [125] Juncheng Shen, Xiaolei Zhu, and De Ma. Tensorclog: An imperceptible poisoning attack on deep neural network applications. *IEEE Access*, 7:41498–41506, 2019.
- [126] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [127] Prabhishkek Singh and Ramneet Singh Chadha. A survey of digital watermarking techniques, applications and attacks. *International Journal of Engineering and Innovative Technology (IJEIT)*, 2(9):165–175, 2013.
- [128] David Solans, Battista Biggio, and Carlos Castillo. Poisoning attacks on algorithmic fairness. *arXiv preprint arXiv:2004.07401*, 2020.
- [129] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlece Fernandes, Bo Li, Amir Rahmati, Florian Tramèr, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18)*, 2018.
- [130] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 2012.
- [131] Otilia Steadman. Her colleagues watched her onlyfans account at work. when bosses found out, they fired her. *BuzzFeedNews*.
- [132] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014.
- [133] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2892–2900, 2015.

- [134] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016.
- [135] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, 2015.
- [136] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.
- [137] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.
- [138] Yaniv Taigman, Ming Yang, Marc’ Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.
- [139] The YouTube Team. Protecting our extended workforce and the community. <https://blog.youtube/news-and-events/protecting-our-extended-workforce-and>, March 2020. Online; accessed 29 October 2020.
- [140] Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.
- [141] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *USENIX Security*, 2016.
- [142] Matthew Turk and Alex Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition*, pages 586–587, 1991.
- [143] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019.
- [144] Vijaya and Matt Derella. An update on our continuity strategy during covid-19. [https://blog.twitter.com/en\\_us/topics/company/2020/An-update-on-our-continuity-strategy-during-COVID-19.html](https://blog.twitter.com/en_us/topics/company/2020/An-update-on-our-continuity-strategy-during-COVID-19.html), March 2020. Online; accessed 29 October 2020.

- [145] Sarah Theres Völkel, Renate Haeuslschmid, Anna Werner, Heinrich Hussmann, and Andreas Butz. How to trick ai: Users' strategies for protecting themselves from automatic personality assessment. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2020.
- [146] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018.
- [147] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8684–8694, 2020.
- [148] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.
- [149] Eric Wong and J Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. *arXiv preprint arXiv:1711.00851*, 2017.
- [150] Can Xiang, Chunming Tang, Yunlu Cai, and Qiuxia Xu. Privacy-preserving face recognition with outsourced computation. *Soft Computing*, 20(9):3735–3744, 2016.
- [151] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. *arXiv preprint arXiv:1703.08603*, 2017.
- [152] Vivek Yadav. p2-traffic signs. <https://github.com/vxy10/p2-TrafficSigns>, 2016.
- [153] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.
- [154] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.
- [155] Fangyi Zhang, Jürgen Leitner, Michael Milford, Ben Upcroft, and Peter Corke. Towards vision-based deep reinforcement learning for robotic motion control. *arXiv preprint arXiv:1511.03791*, 2015.

- [156] Chen Zhu, W Ronny Huang, Ali Shafahi, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. *arXiv preprint arXiv:1905.05897*, 2019.

Appendix A

**ADDITIONAL TABLES AND FIGURES FOR ROBUST  
PHYSICAL-WORLD PERTURBATIONS**

Table A.1: Targeted physical perturbation experiment results on LISA-CNN using a poster-printed Stop sign (subtle attacks) and a real Stop sign (camouflage graffiti attacks, camouflage art attacks). For each image, the top two labels and their associated confidence values are shown. The misclassification target was Speed Limit 45. See Table 3.1 for example images of each attack. Legend: SL45 = Speed Limit 45, STP = Stop, YLD = Yield, ADL = Added Lane, SA = Signal Ahead, LE = Lane Ends.

Distance & Angle	Poster-Printing			Sticker		
	Subtle		Camouflage–Graffiti	Camouflage–Art		
5' 0°	SL45 (0.86)	ADL (0.03)	STP (0.40)	SL45 (0.27)	SL45 (0.64)	LE (0.11)
5' 15°	SL45 (0.86)	ADL (0.02)	STP (0.40)	YLD (0.26)	SL45 (0.39)	STP (0.30)
5' 30°	SL45 (0.57)	STP (0.18)	SL45 (0.25)	SA (0.18)	SL45 (0.43)	STP (0.29)
5' 45°	SL45 (0.80)	STP (0.09)	YLD (0.21)	STP (0.20)	SL45 (0.37)	STP (0.31)
5' 60°	SL45 (0.61)	STP (0.19)	STP (0.39)	YLD (0.19)	SL45 (0.53)	STP (0.16)
10' 0°	SL45 (0.86)	ADL (0.02)	SL45 (0.48)	STP (0.23)	SL45 (0.77)	LE (0.04)
10' 15°	SL45 (0.90)	STP (0.02)	SL45 (0.58)	STP (0.21)	SL45 (0.71)	STP (0.08)
10' 30°	SL45 (0.93)	STP (0.01)	STP (0.34)	SL45 (0.26)	SL45 (0.47)	STP (0.30)
15' 0°	SL45 (0.81)	LE (0.05)	SL45 (0.54)	STP (0.22)	SL45 (0.79)	STP (0.05)
15' 15°	SL45 (0.92)	ADL (0.01)	SL45 (0.67)	STP (0.15)	SL45 (0.79)	STP (0.06)
20' 0°	SL45 (0.83)	ADL (0.03)	SL45 (0.62)	STP (0.18)	SL45 (0.68)	STP (0.12)
20' 15°	SL45 (0.88)	STP (0.02)	SL45 (0.70)	STP (0.08)	SL45 (0.67)	STP (0.11)
25' 0°	SL45 (0.76)	STP (0.04)	SL45 (0.58)	STP (0.17)	SL45 (0.67)	STP (0.08)
30' 0°	SL45 (0.71)	STP (0.07)	SL45 (0.60)	STP (0.19)	SL45 (0.76)	STP (0.10)
40' 0°	SL45 (0.78)	LE (0.04)	SL45 (0.54)	STP (0.21)	SL45 (0.68)	STP (0.14)



Table A.2: Poster-printed perturbation (faded arrow) attack against the LISA-CNN for a Right Turn sign at varying distances and angles. See example images in Table 1 of the main text. Our targeted-attack success rate is 73.33%.

Distance & Angle	Top Class (Confid.)	Second Class (Confid.)
5' 0°	Stop (0.39)	Speed Limit 45 (0.10)
5' 15°	Yield (0.20)	Stop (0.18)
5' 30°	Stop (0.13)	Yield (0.13)
5' 45°	Stop (0.25)	Yield (0.18)
5' 60°	Added Lane (0.15)	Stop (0.13)
10' 0°	Stop (0.29)	Added Lane (0.16)
10' 15°	Stop (0.43)	Added Lane (0.09)
10' 30°	Added Lane (0.19)	Speed limit 45 (0.16)
15' 0°	Stop (0.33)	Added Lane (0.19)
15' 15°	Stop (0.52)	Right Turn (0.08)
20' 0°	Stop (0.39)	Added Lane (0.15)
20' 15°	Stop (0.38)	Right Turn (0.11)
25' 0°	Stop (0.23)	Added Lane (0.12)
30' 0°	Stop (0.23)	Added Lane (0.15)
40' 0°	Added Lane (0.18)	Stop (0.16)

Table A.3: Drive-by testing summary for LISA-CNN. In our baseline test, all frames were correctly classified as a Stop sign. We have added the yellow boxes as a visual guide manually.



Perturbation	Attack Success	A Subset of Sampled Frames $k = 10$
Subtle poster	100%	
Camouflage abstract art	84.8%	

Table A.4: A camouflage art attack on GTSRB-CNN. See example images in Table 3.1. The targeted-attack success rate is 80% (true class label: Stop, target: Speed Limit 80).

Distance & Angle	Top Class (Confid.)	Second Class (Confid.)
5' 0°	Speed Limit 80 (0.88)	Speed Limit 70 (0.07)
5' 15°	Speed Limit 80 (0.94)	Stop (0.03)
5' 30°	Speed Limit 80 (0.86)	Keep Right (0.03)
5' 45°	<b>Keep Right</b> (0.82)	Speed Limit 80 (0.12)
5' 60°	Speed Limit 80 (0.55)	Stop (0.31)
10' 0°	Speed Limit 80 (0.98)	Speed Limit 100 (0.006)
10' 15°	<b>Stop</b> (0.75)	Speed Limit 80 (0.20)
10' 30°	Speed Limit 80 (0.77)	Speed Limit 100 (0.11)
15' 0°	Speed Limit 80 (0.98)	Speed Limit 100 (0.01)
15' 15°	<b>Stop</b> (0.90)	Speed Limit 80 (0.06)
20' 0°	Speed Limit 80 (0.95)	Speed Limit 100 (0.03)
20' 15°	Speed Limit 80 (0.97)	Speed Limit 100 (0.01)
25' 0°	Speed Limit 80 (0.99)	Speed Limit 70 (0.0008)
30' 0°	Speed Limit 80 (0.99)	Speed Limit 100 (0.002)
40' 0°	Speed Limit 80 (0.99)	Speed Limit 100 (0.002)

Table A.5: Sticker perturbation attack on the Inception-v3 classifier. The original classification is microwave and the attacker’s target is phone. See example images in Table A.7. Our targeted-attack success rate is 90%

Distance & Angle	Top Class (Confid.)	Second Class (Confid.)
2' 0°	Phone (0.78)	Microwave (0.03)
2' 15°	Phone (0.60)	Microwave (0.11)
5' 0°	Phone (0.71)	Microwave (0.07)
5' 15°	Phone (0.53)	Microwave (0.25)
7' 0°	Phone (0.47)	Microwave (0.26)
7' 15°	Phone (0.59)	Microwave (0.18)
10' 0°	Phone (0.70)	Microwave (0.09)
10' 15°	Phone (0.43)	Microwave (0.28)
15' 0°	<b>Microwave (0.36)</b>	Phone (0.20)
20' 0°	Phone (0.31)	Microwave (0.10)

Table A.6: Sticker perturbation attack on the Inception-v3 classifier. The original classification is coffee mug and the attacker’s target is cash machine. See example images in Table A.8. Our targeted-attack success rate is 71.4%.

Distance & Angle	Top Class (Confid.)	Second Class (Confid.)
8” 0°	Cash Machine (0.53)	Pitcher (0.33)
8” 15°	Cash Machine (0.94)	Vase (0.04)
12” 0°	Cash Machine (0.66)	Pitcher (0.25)
12” 15°	Cash Machine (0.99)	Vase (<0.01)
16” 0°	Cash Machine (0.62)	Pitcher (0.28)
16” 15°	Cash Machine (0.94)	Vase (0.01)
20” 0°	Cash Machine (0.84)	Pitcher (0.09)
20” 15°	Cash Machine (0.42)	Pitcher (0.38)
24” 0°	Cash Machine (0.70)	Pitcher (0.20)
24” 15°	<b>Pitcher (0.38)</b>	Water Jug (0.18)
28” 0°	<b>Pitcher (0.59)</b>	Cash Machine (0.09)
28” 15°	Cash Machine (0.23)	Pitcher (0.20)
32” 0°	<b>Pitcher (0.50)</b>	Cash Machine (0.15)
32” 15°	<b>Pitcher (0.27)</b>	Mug (0.14)

Table A.7: Uncropped images of the microwave with an adversarial sticker designed for Inception-v3.









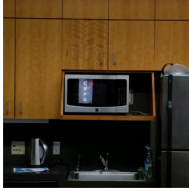
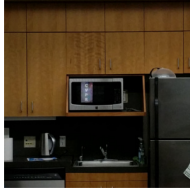
Distance/Angle	Image	Distance/Angle	Image
2' 0°		2' 15°	
5' 0°		5' 15°	
7' 0°		7' 15°	
10' 0°		10' 15°	
15' 0°		20' 0°	

Table A.8: Cropped Images of the coffee mug with an adversarial sticker designed for Inception-v3.

Distance/Angle	Image	Distance/Angle	Image
8"0°		8"15°	
12"0°		12"15°	
16"0°		16"15°	
20"0°		20"15°	
24"0°		24"15°	
28"0°		28"15°	
32"0°		32"15°	

## Appendix B

### ADDITIONAL MATERIAL FOR FOGGYSIGHT

#### ***B.1 Adversarial Examples Success***

In this appendix, we analyze how well the adversarial examples generation algorithm achieves its goal of shifting the output of the neural network while producing images indistinguishable from the original photo. To do so, we begin by measuring the final distance in the embedding space between the vectors produced by the neural network for decoy photos and their respective targets. The results are given in Fig. B.1.

As expected, we can observe that all perturbation amounts manage to shift the output of the neural network. Furthermore, higher perturbation amounts are more successful at bringing the final neural network loss close to their target. Note that even at the highest perturbation amounts, there is a level of “irreducible” loss and the optimization algorithm does not always achieve its goal perfectly. It is also useful to understand how these perturbations look visually. We show the final decoy images with different perturbation amounts in Fig. B.2. Even high perturbation amounts do not distort the image to an unrecognizable amount. Therefore, we do not believe that this will have a high impact on user experience.

#### ***B.2 Alternative Targeting Mechanisms***

We also analyzed alternative targeting mechanisms in addition to those presented in Section 4.4.3 and present the results from those experiments here.

**Same Universal Target.** We begin with the strategy of selecting the same single photo of the protected to serve as a target for the decoys of all protectors. The results are given in Fig. B.3. While this is the simplest strategy that exposes the least information about the protected to the protectors, these benefits come at a large cost. We can observe that recall is only moderately impacted (an ideal



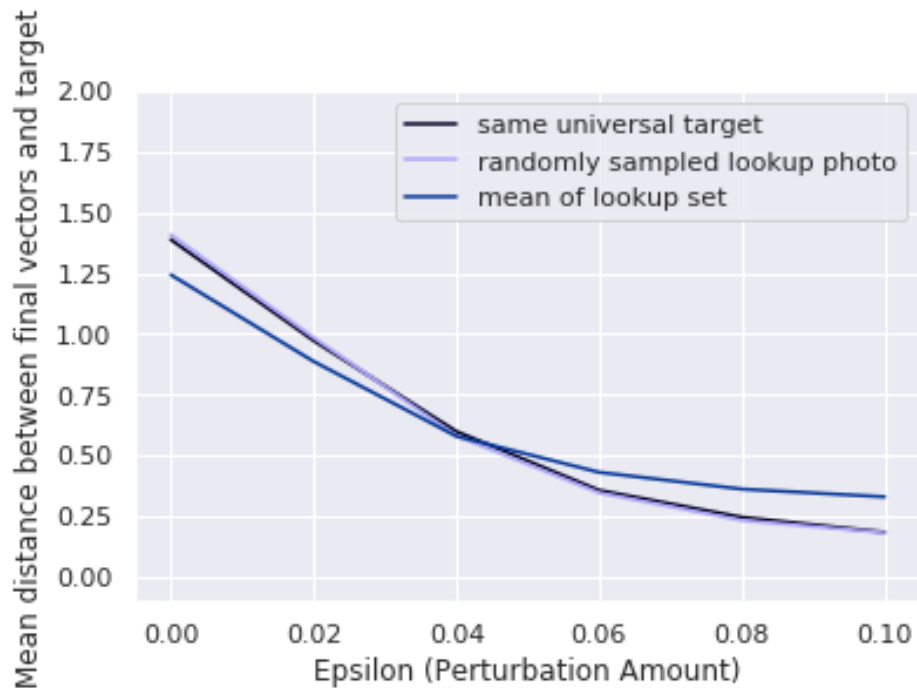


Figure B.1: Magnitude of final optimization loss after decoy photo generation under different perturbation magnitudes  $\epsilon$ . Note that the case where  $\epsilon = 0.0$  corresponds to the unmodified photos. As expected, the higher the perturbation amount, the better the PGD algorithm for adversarial examples generation achieves its goal.

protection scheme brings recall down to 0.0). In fact, even at high perturbation magnitudes, a photo with the real identity of the protected is the closest neighbor to the query between 80% and 90% of the time. (See Fig. B.3a and the values for recall at  $k = 1$ . When the recall set contains only one photo, that photo is the closest neighbor to the query.) The discovery rate remains consistently high for all perturbation amounts and recall set sizes, which indicates that at least one photo of the protected is available in a high percentage of the searches ( $> 90\%$ ).

**Targeting a Sample from a Gaussian Model.** As another alternative, we evaluate targeting a *sample* from a Gaussian model with mean and standard deviation matching that of the lookup set. Results are given in Fig. B.4. The results at all settings of  $k$  and  $\epsilon$  are as good or worse

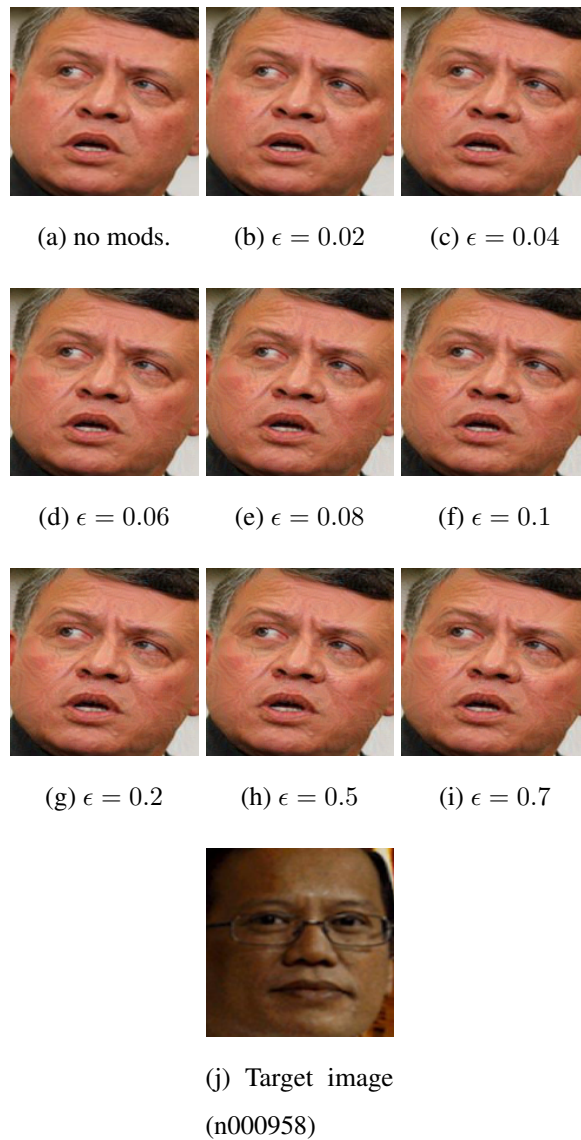
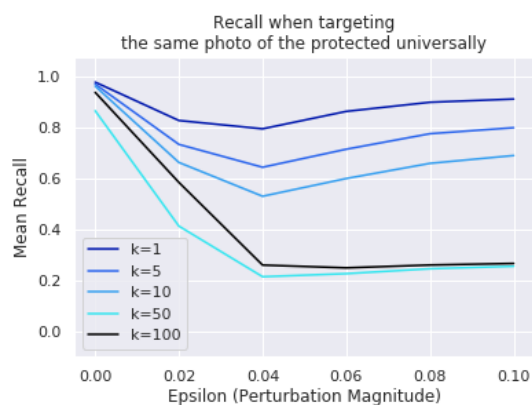
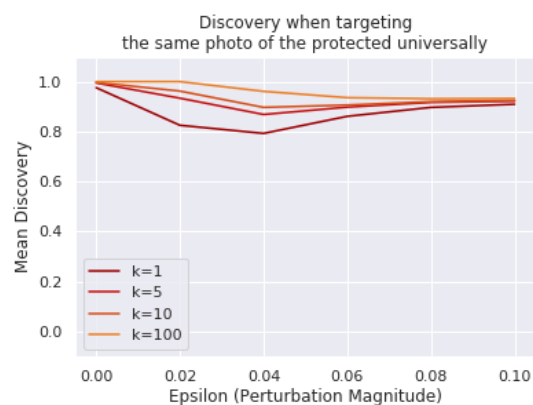


Figure B.2: Illustration of final decoy images under different perturbation magnitudes  $\epsilon$ . These are images of subject n000029 in the VGGFace2 dataset modified according to the “randomly sampled target from the lookup set” strategy to produce vectors in the region of subject n000958.

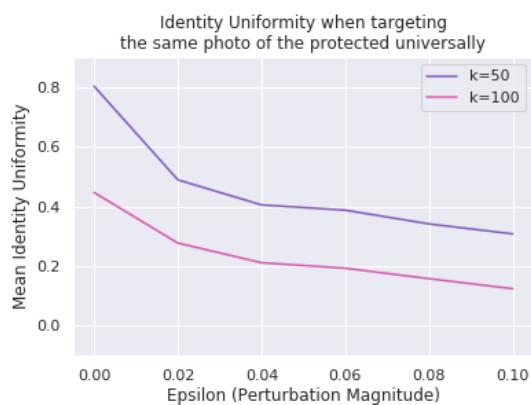
than the results when targeting the mean. For example, for  $\epsilon = 0.06$  and  $k = 5$ , discovery (in Fig. B.4b) remains up to 10% higher. This is likely because the residual loss from not achieving



(a) Recall when targeting the same photo of the protected user



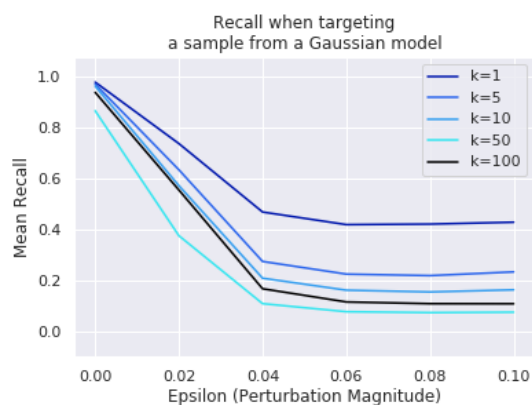
(b) Discovery when targeting the same photo of the protected user



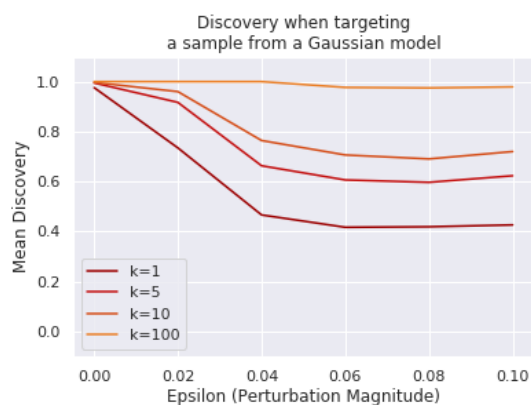
(c) Identity uniformity when targeting the same photo of the protected user

Figure B.3: Privacy strategy success when targeting the same photo of the protected user universally. All results averaged over all identities and all photos. While this strategy does manage to bring recall down, it is less effective at reducing the discovery rate and the uniformity of identities in the top recall set.

the optimization objective perfectly introduces enough variation when targeting the mean to scatter the decoys well. By contrast, when we purposefully introduce additional error through targeting a sample from a non-0 variance Gaussian, the decoys land farther away from the query photo.



(a) Recall when targeting a sample from a Gaussian model



(b) Discovery when targeting a sample from a Gaussian model



(c) Identity uniformity when targeting a sample from a Gaussian model

Figure B.4: Graphs of privacy strategy success when targeting a sample from a Gaussian model. Observe that this scheme fares just as well as when targeting the mean lookup set by comparing with Figure 4.3.

### B.3 Solo Action Defenses with Untargeted Adversarial Examples

#### B.3.1 Setup and Motivation

Here, we consider the most natural strategy for an individual with identity  $i$  trying to protect their own privacy while acting alone. Recall that the face recognition pipeline has a dataset of lookup

photos  $L$ . Those photos in  $L_i \subset L$  depict identity  $i$ , and correspond to photos of individual  $i$  from social media websites. To protect their privacy, individual  $i$  aims to modify the photos  $x_i \in L_i$  such that  $D(x_i, q_i)$  is large for some future query  $q_i$ . The individual must modify their photos prior to those photos being scraped by the face recognition system; this is a key issue that we discuss in more detail later. Unfortunately, the individual cannot predict future query photos  $q_i$ . However, future query photos will by definition be close to the unmodified  $x_i \in L_i$ . Thus, it is natural to instead modify  $x_i$  to be far away from itself. We do this by solving the following optimization problem:

$$A(x_i) = \arg \max_z D(x_i, z) \text{ such that } \|x_i - z\|_\infty \leq \epsilon$$

where  $x_i$  is the image that depicts individual  $i$  — one that the individual is potentially trying to upload to social media —  $A$  is the adversarial modification that transforms  $x_i$ , and  $\epsilon$  is a pre-defined perturbation amount. This attack aims to make sure that, to the network,  $x_i$  is not recognizable as the identity of the individual depicted in it, while maintaining via the constraint that it appears like a normal photo to a human observer.

To optimize this function, we use projected gradient descent, which was introduced in the context of adversarial examples by [88]. Although usually adversarial examples are initialized as the target image  $z_0 = x_i$  doing so results in the optimization getting stuck  $D(z_0, x_i) = 0$ . We therefore follow the strategy outlined by [88] and initialize the attack with a small amount of random noise  $z_0 = x_i + \hat{\mathcal{N}}_\epsilon(0, \sigma)$  for the truncated normal distribution  $\hat{\mathcal{N}}_\epsilon$  truncated at  $[-\epsilon, \epsilon]$ .

The motivation behind this attack is presented in Fig. B.5. By maximizing the distance in embedding space to the original, clean lookup photos, the target minimizes the chance that a new, clean photo will match any of the modified lookup set photos.

### B.3.2 Experimental Evaluation

The result of applying the self distance attack to all photos belonging to the user in the database is shown in Fig. B.6. The graph averages the recall percentage and discovery rate over all images and all identities in the lookup set. The chart shows that perturbation amount of 0.04 relative to an

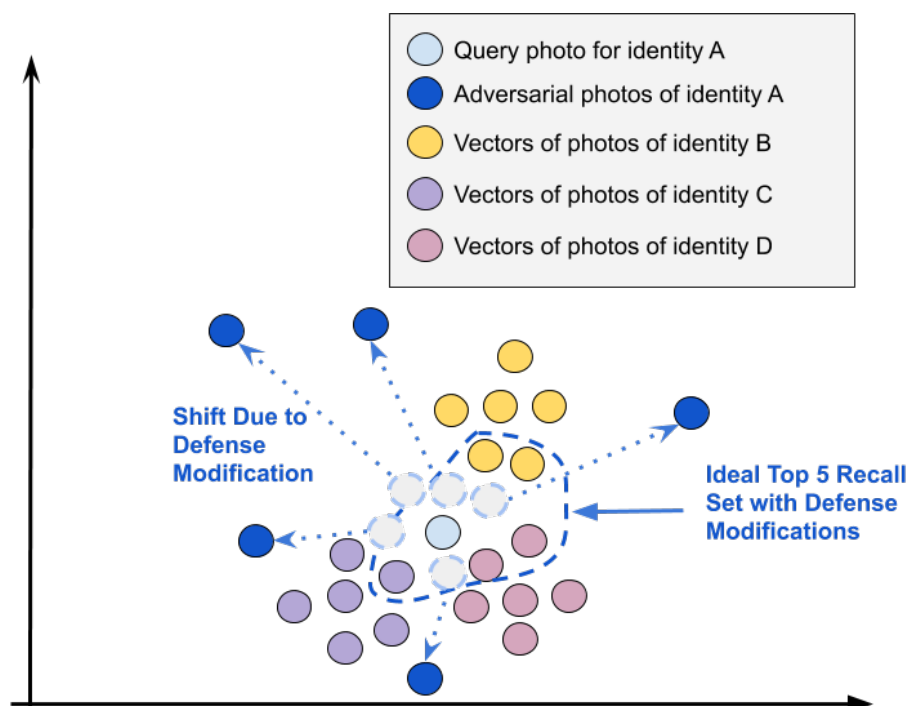
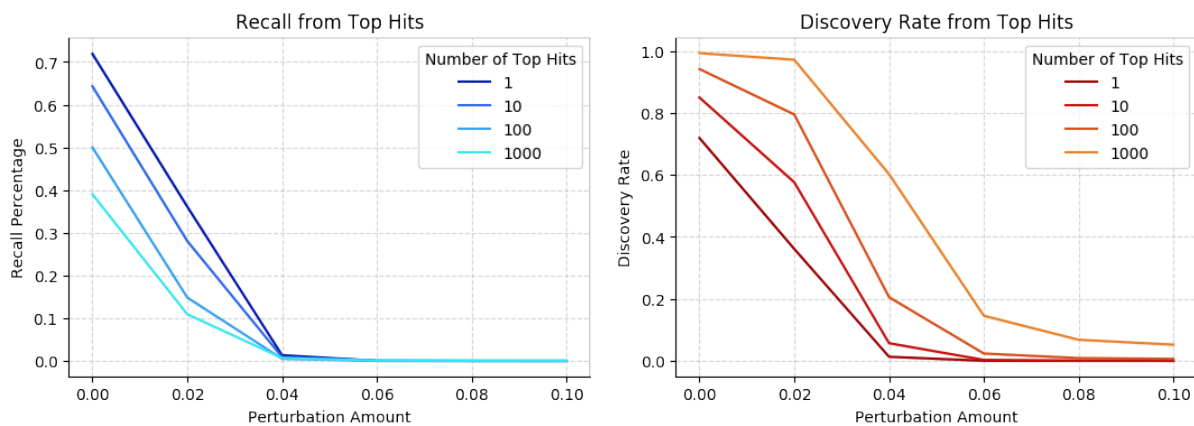


Figure B.5: A visual illustration of the solo action defense. A user aims to shift his or her face images far away from their original location in the embedding space. This fills the recall set with other identities.

image standard deviation of 1 suffices the drop the recall percentage to almost 0 and the discovery rate at  $k = 100$  to approximately 10%.

However, it is not always reasonable to assume that users control 100% of their photos in the database. Therefore, we next study the performance of the solo action attacks if only the target can only modify some fraction of their photos in the lookup set. We define the *subsample rate* as the percentage of the target's photos in the lookup set that the target can modify. That is, if the adversary has 100 photos of the target in their lookup set, and the target can modify 70 of them, then the subsample rate is 70%. We plot the result of subsampling using the self distance and target pair strategies in Fig. B.7. The plots show that subsampling even at a rate of 75% drastically increases the expected discovery rate, which indicates that face recognition systems need only a few photos of a target out of hundreds of thousands in order to identify them. This indicates that our proposed

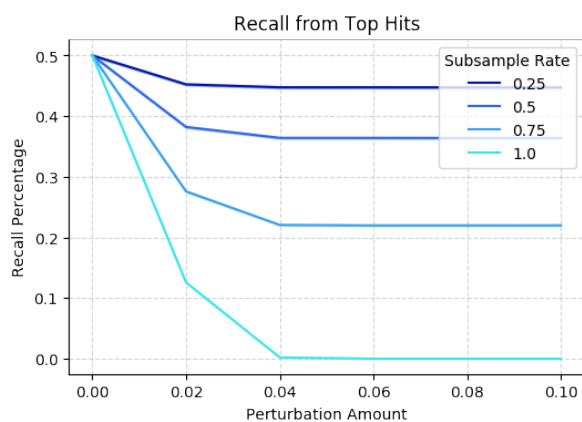


(a) Recall for the solo action attack

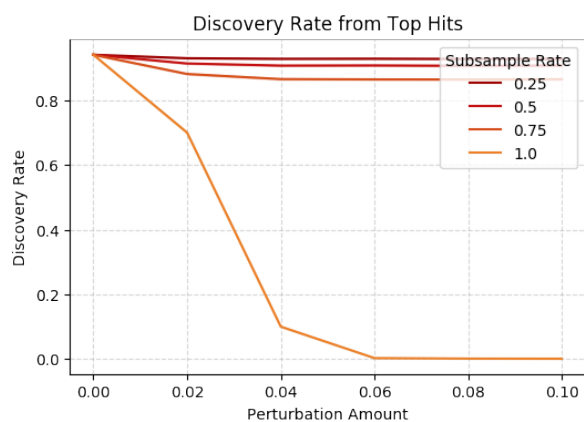
(b) Discovery rate for the solo action attack

Figure B.6: Recall and discovery rate at various levels of  $k$  and  $\epsilon$  when assuming the protected has 100% control of their own lookup set. The perturbation amount is normalized to represent percentage relative to standard deviation (images have unit standard deviation). For both metrics, a perturbation amount of 0.04 suffices to evade recognition. “Top Hits” refers to the recall set of nearest neighbors to the query photo that is returned by the facial search service to its user.

attacks may not be effective enough in the case that the adversary has photos of the protected that the protected cannot modify. In this case, different strategies that involve many protectors acting in coordination may be needed.



(a) Recall for the solo action attack with limited control of the lookup set



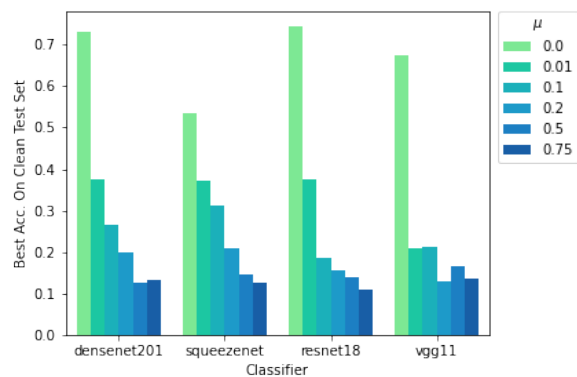
(b) Discovery rate for the solo action attack with limited control of the lookup set

Figure B.7: Recall and discovery rate at various levels of  $k$  and  $\epsilon$  when assuming the protected only has limited control of their own lookup set (as controlled by the subsample rate). The perturbation amount is normalized to represent percentage relative to standard deviation (images are have unit standard deviation). Only having access to a fraction of the lookup data drastically degrades privacy protection. This indicates that other strategies are needed in the case that we cannot modify 100% of the target's data. "Top Hits" refers to the recall set of nearest neighbors to the query photo that is returned by the facial search service to its user.

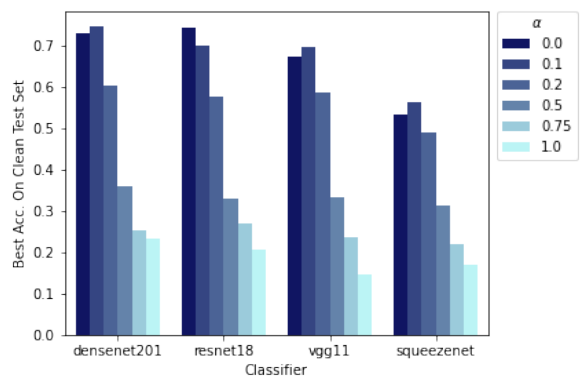


Appendix C

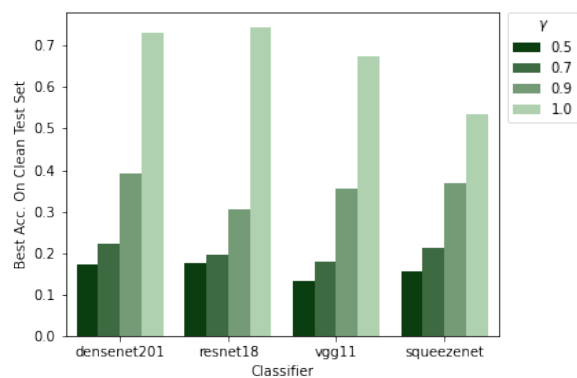
**ADDITIONAL FIGURES FOR DISRUPTING UNAUTHORIZED USES  
OF MACHINE LEARNING**



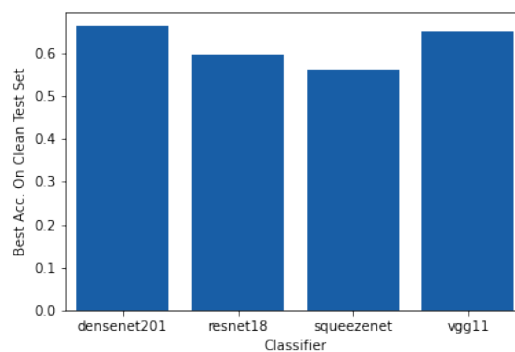
(a) Best achievable accuracy after 50 epochs across a range of hyperparameters and random seeds for the pixel-based disruption approach.  $\mu = 0.0$  corresponds to the unmodified, clean dataset for this approach.



(b) Best achievable accuracy after 50 epochs across a range of hyperparameters and random seeds for the visible watermark disruption approach.  $\alpha = 0.0$  corresponds to the unmodified, clean dataset for this approach.

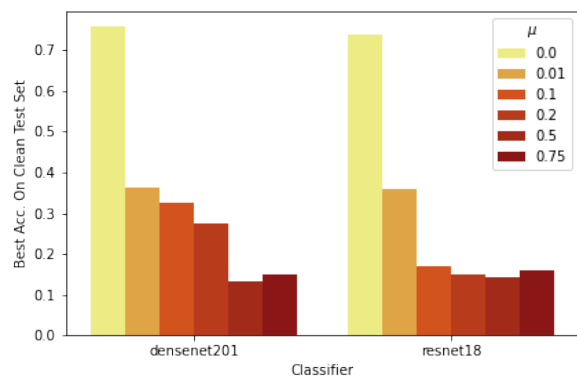


(c) Best achievable accuracy after 50 epochs across a range of hyperparameters and random seeds for the brightness modulation pattern disruption approach.  $\gamma = 1.0$  corresponds to the unmodified, clean dataset for this approach.

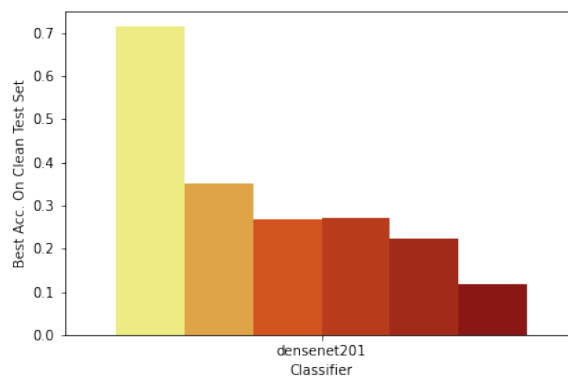


(d) Best achievable accuracy after 50 epochs across a range of hyperparameters and random seeds for the Fowl et al. [37] approach

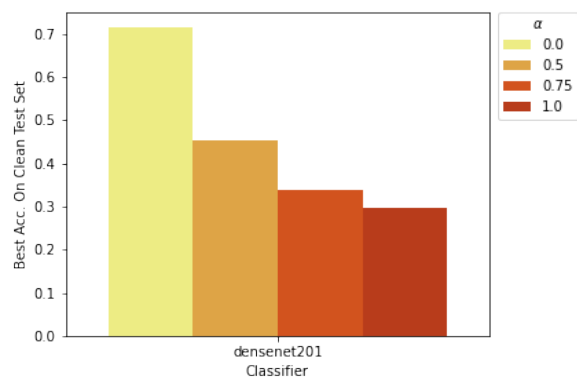
Figure C.1: Results on training a modified CIFAR10 with no countermeasures.



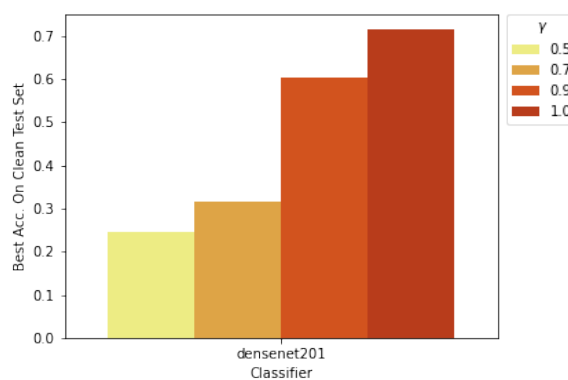
(a) Best achievable accuracy after 50 epochs across a range of hyperparameters and random seeds for the pixel-based disruption approach after applying Gaussian noise to the images.  $\mu = 0.0$  corresponds to the unmodified, clean dataset for this approach.



(b) Best achievable accuracy after 50 epochs across a range of hyperparameters and random seeds for the pixel-based disruption approach after applying aggressive training-time augmentations.  $\mu = 0.0$  corresponds to the unmodified, clean dataset for this approach.



(c) Best achievable accuracy after 50 epochs across a range of hyperparameters and random seeds for the visible watermark disruption approach after applying aggressive training-time augmentations.  $\alpha = 0.0$  corresponds to the unmodified, clean dataset for this approach.



(d) Best achievable accuracy after 50 epochs across a range of hyperparameters and random seeds for the brightness modulation patterns disruption approach after applying aggressive training-time augmentations.  $\gamma = 1.0$  corresponds to the unmodified, clean dataset for this approach.

Figure C.2: Results when applying countermeasures to the modified CIFAR10 training set.

Figure C.3: Examples of the perturbed CIFAR10 training set with a pixel-based perturbation approach at various settings of the parameter  $\mu$ .



Figure C.4: Examples of the perturbed CIFAR10 training set with a visual watermarking approach at various settings of the parameter  $\alpha$ .

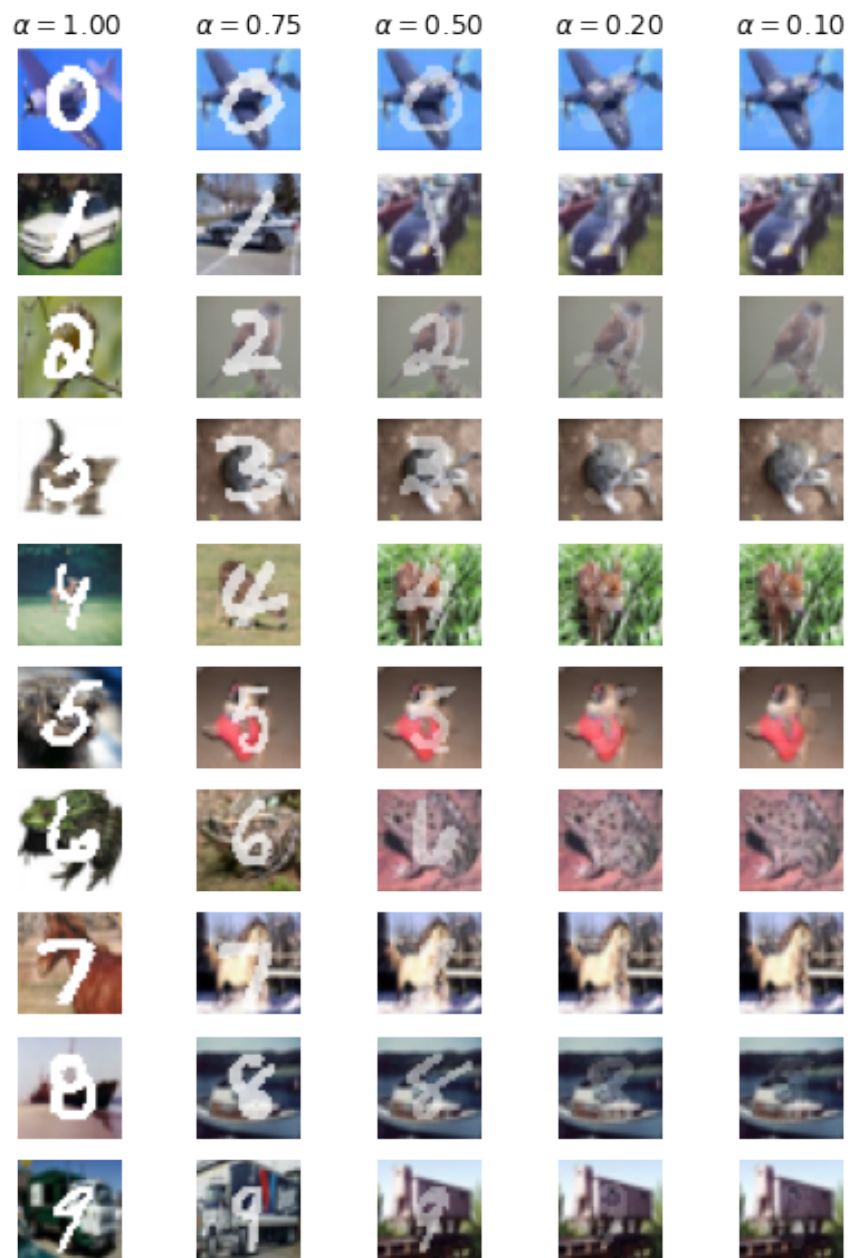


Figure C.5: Examples of the perturbed CIFAR10 training set with a brightness modulation approach at various settings of the parameter  $\gamma$ .

