ABSTRACT

Title of Dissertation: ADVERSARIAL ROBUSTNESS AND FAIRNESS IN DEEP LEARNING

Valeriia Cherepanova
Doctor of Philosophy, 2023

Dissertation Directed by: Tom Goldstein
Department of Computer Science

While deep learning has led to remarkable advancements across various domains, the widespread adoption of neural network models has brought forth significant challenges such as vulnerability to adversarial attacks and model unfairness. These challenges have profound implications for privacy, security, and societal impact, requiring thorough investigation and development of effective mitigation strategies.

In this work we address both these challenges. We study adversarial robustness of deep learning models and explore defense mechanisms against poisoning attacks. We also explore the sources of algorithmic bias and evaluate existing bias mitigation strategies in neural networks. Through this work, we aim to contribute to the understanding and enhancement of both adversarial robustness and fairness of deep learning systems.

ADVERSARIAL ROBUSTNESS AND FAIRNESS IN DEEP LEARNING

by

Valeriia Cherepanova

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2023

Advisory Committee:
Professor Tom Goldstein, Chair/Advisor
Professor John Dickerson
Professor Furong Huang
Professor Hal Daume III
Professor Daniel Butts

Acknowledgments

First and foremost, I would like to thank my advisor, Professor Tom Goldstein, for his guidance and mentorship, which have been foundational for my academic and personal growth. Tom showed me through his own example how to set high research standards and pursue "crazy ideas" at the same time. I am thankful to Tom for his help with shaping research directions, his invaluable insights, and for giving me the freedom to explore various research topics and directions throughout my PhD.

Micah Goldblum has played an important role in my development as a scientist. Not only did he introduce me to the world of deep learning research, but he also supported me on every project, as a mentor, collaborator, and a friend.

Throughout my PhD I worked closely with amazing students, researchers and professors including John Dickerson, Andrew Gordon Wilson, Bayan Bruss, Micah Goldblum, Jonas Geiping, Steven Reich, Arpit Bansal, Roman Levin, Avi Schwarzschild, Eitan Borgnia, Liam Fowl, Vedant Nanda, Samuel Dooley, Gowthami Somepalli, Amin Ghiasi and many others. I am deeply grateful for their help on the projects, continuous support and for their impact on my professional growth.

Finally, I owe a big thank you to my family. My parents invested a lot of effort in my education and made it possible for me to come to the US for graduate study. They have been incredibly supportive of all my decisions, even at times when I was unable to see my family for

years because of my commitments. Finally, none of this would have been possible without my husband and co-author, Roman Levin. He has always been there for me, supporting me through deadlines, providing research advice whenever I needed help, and standing by my side through all my ups and downs.

Table of Contents

Acknowledgements ii

Table of Contents iv

List of Tables vii

List of Figures xi

Chapter 1: Introduction 1

Chapter 2: Preliminaries 4
- 2.1 Neural Networks . . . 4
- 2.2 Face Recognition . . . 6

Chapter 3: LowKey: Leveraging Adversarial Attacks to Protect Social Media Users from Facial Recognition 9
- 3.1 Introduction . . . 10
- 3.2 Related Work . . . 11
- 3.3 The LowKey Attack on Mass Surveillance . . . 13
  - 3.3.1 Problem Setup . . . 13
  - 3.3.2 The LowKey attack . . . 15
- 3.4 Experimental Design . . . 17
- 3.5 Breaking Commercial Black-Box APIs . . . 18
- 3.6 Additional Experiments . . . 19
  - 3.6.1 Ensembles and Transferability . . . 20
  - 3.6.2 Gaussian Smoothing . . . 21
  - 3.6.3 Run-Time . . . 22
  - 3.6.4 Robustness to Image Compression . . . 23
  - 3.6.5 Scalability to All Image Sizes (Disclaimer) . . . 24
- 3.7 Discussion . . . 26
- 3.8 Experimental Details and Additional Results . . . 27
  - 3.8.1 Implementation Details . . . 27
  - 3.8.2 Rank-1 accuracy on FaceScrub data . . . 27
  - 3.8.3 Results on UMDFaces dataset . . . 27
  - 3.8.4 Can we reduce the size of our attack? . . . 28
  - 3.8.5 Comparison with Fawkes . . . 29
  - 3.8.6 Gaussian Smoothing in LowKey . . . 29

Chapter 4: Strong Data Augmentation Sanitizes Poisoning and Backdoor Attacks Without an Accuracy Tradeoff 33
4.1 Introduction . . . 34
4.2 Threat Model . . . 36
4.3 Method . . . 38
4.4 Experiments . . . 40
4.4.1 Defending Against Backdoor Trigger Attacks . . . 40
4.4.2 Defending Against Targeted Poisoning Attacks . . . 41
4.5 Conclusions . . . 43

Chapter 5: A Deep Dive into Dataset Imbalance and Bias in Face Identification 45
5.1 Introduction . . . 46
5.2 Related Work . . . 49
5.2.1 Imbalance in verification . . . 49
5.2.2 Bias in Identification . . . 50
5.2.3 Imbalance in Deep Learning . . . 51
5.2.4 Other sources of bias in facial recognition . . . 51
5.3 Face Identification Setup . . . 52
5.4 Balance in the Train Set . . . 54
5.4.1 Balancing the number of identities . . . 54
5.4.2 Balancing the number of images per identity . . . 56
5.5 Balance in the Test Set . . . 57
5.5.1 Balancing the number of identities . . . 57
5.5.2 Balancing the number of images per identity . . . 58
5.6 A cautionary tale: matching the balance in the train and gallery data . . . 60
5.7 Bias comparisons . . . 61
5.7.1 Bias in random feature extractors . . . 61
5.7.2 Are models biased like humans? . . . 62
5.8 Actionable Insights . . . 64
5.9 Experimental Details and Additional Results . . . 66
5.9.1 Results for models trained without class-balanced sampling. . . . 66
5.9.2 Additional Plots and Tables . . . 66

Chapter 6: Technical Challenges for Training Fair Neural Networks 76
6.1 Introduction . . . 76
6.2 Related work . . . 79
6.3 Experimental setup . . . 82
6.3.1 Face recognition . . . 82
6.3.2 Medical image classification . . . 83
6.4 Fairness notions . . . 84
6.4.1 Training with fairness regularization . . . 87
6.4.2 Imposing fairness constraints on a holdout set . . . 90
6.4.3 Max-Min training . . . 91
6.4.4 Adjusted angular margins for face recognition . . . 91
6.5 Fair feature representations do not yield fair model behavior . . . 92

6.6 A simple baseline for fairness: label flipping 94
6.7 Fairness gerrymandering 96
6.8 Discussion 98
6.9 Experimental Details and Additional Results 100
6.9.1 Medical Image Classification - CheXpert 100
6.9.2 Face Recognition 101
6.10 Additional Results 103

Bibliography 108

List of Tables

3.1 An evaluation of Amazon Rekognition and Microsoft Azure Face on FaceScrub data with LowKey and Fawkes protection (a small number, and lighter color, indicates a successful attack). LowKey consistently achieves virtually flawless protection, while Fawkes provides little protection. 18

3.2 Rank-50 accuracy of the LowKey and Fawkes attacks. After the first two rows, each row represents LowKey attacks generated from the same model. Each column represents inference on a single model. The first two letters in the model's name denote the type of backbone: IR or ResNet (RN). The last letter in the model's name indicates the type of head; "A" denotes ArcFace, and "C" denotes CosFace. Smaller numbers, and lighter colors, indicate more successful attacks. . 21

3.3 Rank-50 accuracy of FR models tested on blurred LowKey images computed with/without Gaussian smoothing. 22

3.4 Evaluation of LowKey on full-size images. Rows indicate levels of magnitude of LowKey (denoted by the number of attack steps). 26

3.5 Rank-1 accuracy of the LowKey and Fawkes attacks on the FaceScrub dataset. After the first two rows, each row represents LowKey attacks generated from the same model. Each column represents inference on a single model. The first two letters in the model's name denote the type of backbone: IR or ResNet (RN). The last letter in the model's name indicates the type of head; "A" denotes ArcFace, and "C" denotes CosFace. Smaller numbers, and lighter colors, indicate more successful attacks. 28

3.6 Rank-50 accuracy of LowKey attacks on the UMDFaces dataset. After the first row, each row represents LowKey attacks generated from the same model. Each column represents inference on a single model. The first two letters in the model's name denote the type of backbone: IR or ResNet (RN). The last letter in the model's name indicates the type of head; "A" denotes ArcFace, and "C" denotes CosFace. Smaller numbers, and lighter colors, indicate more successful attacks. . 29

3.7 Rank-1 accuracy of LowKey attack attacks on the UMDFaces dataset. After the first row, each row represents LowKey attacks generated from the same model. Each column represents inference on a single model. The first two letters in the model's name denote the type of backbone: IR or ResNet (RN). The last letter in the model's name indicates the type of head; "A" denotes ArcFace, and "C" denotes CosFace. Smaller numbers, and lighter colors, indicate more successful attacks. 30

3.8 Rank-50 accuracy of LowKey small attacks on the UMDFaces dataset. After the first row, each row represents LowKey small attacks generated from the same model. Each column represents inference on a single model. The first two letters in the model's name denote the type of backbone: IR or ResNet (RN). The last letter in the model's name indicates the type of head; "A" denotes ArcFace, and "C" denotes CosFace. Smaller numbers, and lighter colors, indicate more successful attacks. 31
3.9 Rank-1 accuracy of LowKey small attacks on the UMDFaces dataset. After the first row, each row represents LowKey small attacks generated from the same model. Each column represents inference on a single model. The first two letters in the model's name denote the type of backbone: IR or ResNet (RN). The last letter in the model's name indicates the type of head; "A" denotes ArcFace, and "C" denotes CosFace. Smaller numbers, and lighter colors, indicate more successful attacks. 32
3.10 Rank-1 accuracy of FR models tested on blurred images attacked without and with the Gaussian smoothing term. The first two letters in the model's name denote the type of backbone: IR or ResNet (RN). The last letter in the model's name indicates the type of head; "A" denotes ArcFace, and "C" denotes CosFace. Smaller numbers, and lighter colors, indicate more successful attacks. 32

4.1 Poison success and (clean) validation accuracy in the from-scratch setting against backdoor attacks. The first two columns correspond to poisoning all images in the target class, the last two columns correspond to poisoning 10% of images in the target class. All values are averaged over 4 runs. 38
4.2 Poison success and (clean) validation error results for the from-scratch setting against Witches' Brew. All values are averaged over 20 runs. Lower values in the first two columns are better. Higher numbers in the last column are better. 40

5.1 Details on the number of identities, total number of images and average number of images per identity used in experiments with train and test data balance. We also report statistics for the default train and test sets. M denotes male, F denotes female. 54
5.2 **Train Set Id Imbalance.** The female and male accuracy computed over the default balanced test set for models trained on data with various ratios of number of male and female identities. See details of the experiment in Section 5.4.1 . . . 70
5.3 **Train Set Img Imbalance.** The female and male accuracy computed over the default balanced test set for models trained on data with various ratios of number of images per male and female identity. See details of the experiment in Section 5.4.2 71
5.4 **Test Set Id Imbalance.** The female and male accuracy for models trained on default train set computed on test set with various ratios of number of male and female identities. See details of experiment in Section 5.5.1. 72
5.5 **Test Set Img Imbalance.** The female and male accuracy for models trained on default train set computed on test set with various ratios of number of images per male and female identities. See details of the experiment in Section 5.5.2 73

5.6 **Train & Test Set Id Imbalance.** The female and male accuracy for models trained and tested on data with the same ratios of male and female identities. See details of experiment in Section 5.6. 74
5.7 **Train & Test Set Img Imbalance.** The female and male accuracy for models trained and tested on data with the same ratios of number of images per male and female identity. See details of experiment in Section 5.6. 75

6.1 **[CheXpert - training with fairness penalties]** Results for all 5 CheXpert tasks: Cardiomegaly (CA), Edema (ED), Consolidation (CO), Atelectasis (AT) and Pleural Effusion (PE). The regularizer is optimized on the training data, and α here denotes the coefficient of the regularizer. 88
6.2 **[Face Recognition ResNet-18]** Performance of models trained with different training schemes designed for mitigating disparity in misclassification rates between males and females. All models have ResNet-18 backbone and CosFace head. The first column refers to the training scheme used, penalty indicates the size penalty coefficient. The train accuracy is computed in a classification manner, while validation and test accuracies are computed in the 1-nearest neighbors sense. The gap subcolumn refers to the difference between male and female accuracies. 90
6.3 **[Face recognition]** Accuracy and intra- and inter-class angles measured for male and female images for ResNet-152 model trained with **adjusted angle margins**. The numbers are measured on validation and test sets. It can be seen that 'fair' models (trained with increased angle margin for females) improve fairness on validation set, but increase the accuracy gap on test set. 92
6.4 **[Face recognition]** Facial recognition and gender classification test accuracy for ResNet-152 model trained with a sensitive information removal network on top. Here, α denotes the magnitude of adversarial penalty and for sufficiently large α, the discriminator predicts a fixed gender for all images (in bold). Gender in the first column is the gender of images penalized during training. 94
6.5 **[Fairness gerrymandering on BUPT dataset]** The first row shows accuracies obtained with baseline model. The second, third, and fourth blocks reflect performance of models trained with equal loss penalty, adjusted angle margin for females, and randomly flipped labels for males respectively. For the models trained for "gender-fairness", we report differences from baseline. 97
6.6 **[Face Recognition ResNet-18]** Performance of models trained with different training schemes designed for mitigating disparity in misclassification rates between males and females. All models have ResNet-18 backbone and CosFace head. The first column refers to the training scheme used, penalty indicates the size penalty coefficient. The train accuracy is computed in a classification manner, while validation and test accuracies are computed in the 1-nearest neighbors sense. The gap subcolumn refers to the difference between male and female accuracies. For the Fair Features training scheme, "gender" refers to the subgroup of images used in adversarial regularization during training. 104

6.7 **[Face Recognition ResNet-152]** Performance of models trained with different training schemes designed for mitigating disparity in misclassification rates between males and females. All models have ResNet-152 backbone and CosFace head. 105
6.8 **[CheXpert - fairness penalties on the validation set]** Results for all 5 CheXpert tasks: Cardiomegaly (CA), Edema (ED), Consolidation (CO), Atelectasis (AT) and Pleural Effusion (PE). 106
6.9 **[CheXpert - minmax training]** Results for all 5 CheXpert tasks: Cardiomegaly (CA), Edema (ED), Consolidation (CO), Atelectasis (AT) and Pleural Effusion (PE). 106
6.10 **[CheXpert - random label flipping]** Results for all 5 CheXpert tasks: Cardiomegaly (CA), Edema (ED), Consolidation (CO), Atelectasis (AT) and Pleural Effusion (PE). 107

List of Figures

3.1 Top: original images, Bottom: protected by LowKey. 10
3.2 The LowKey pipeline. When users protect their publicly available images with LowKey, facial recognition systems cannot match these harvested images with new images of the user, for example from surveillance cameras. 14
3.3 LowKey attacked images computed without (above) and with (below) Gaussian smoothing. 23
3.4 First row: Original large images, Second row: Images protected with LowKey (medium magnitude), Third row: Images protected with LowKey (large magnitude). 24
3.5 Panel of different attacks. First row: original images, second row: Fawkes attack, third row: LowKey small attack, last row: LowKey attack. 30

4.1 An illustration of CutMix with poisoned data (truck) and non-poisoned data (deer). CutMix chops up the patch, lessening its visual impact in training data. 39
4.2 Mixup averages perturbed and clean images. 40
4.3 Poison success vs. (clean) validation accuracy for non-adaptive attacks (top) and adaptive attacks (bottom). 44

5.1 **Examples of imbalance in face identification**. Top left: data containing more female identities than male identities. Top right: data containing the same number of male and female identities, but more images per male identity. Bottom: two possible test (gallery) sets showing how the effects of different kinds of imbalance may interact. 48
5.2 **Train Set Imbalance.** Results of experiments that change the train set gender presentation balance. Top row: male and female accuracy are plotted against the proportion of male data in the train set. Bottom row: for an alternate view, female accuracy is flipped horizontally, so that it is plotted against the proportion of *female* data in the train set. All models are tested on the default balanced test set. 55
5.3 **Test Set Imbalance.** Results of experiments that change the test set gender presentation balance. Top row: male and female accuracy are plotted against the proportion of male data in the test set. Bottom row: for an alternate view, female accuracy is flipped horizontally, so that it is plotted against the proportion of *female* data in the test set. All models are trained on the default balanced train set. For each experiment, the test set was split with 5 random seeds, and the results are averaged across seeds. 58

5.4 **Train & Test Set Imbalance.** Results of experiments that adjust the gender presentation balance in both the train and test set. Top row: male and female accuracy are plotted against the proportion of male data used in both the train and test set. Bottom row: for an alternate view, female accuracy is flipped horizontally, so that it is plotted against the proportion of *female* data in both the train and test set. For each experiment, the test set was split with 5 random seeds, and the results are averaged across seeds. 59
5.5 **Random Feature Extractors.** The plot illustrates male (blue) and female (orange) accuracy of random feature extractors against the proportion of male images in the test set. The standard deviation is computed across 10 random initializations. 61
5.6 Pearson correlation of $L2$ ratio vs. human accuracy for various models as proportion of male training data varies. 63
5.7 Scatterplots of model L2 ratio vs. human accuracy on each question in the InterRace identification dataset. Both models are MobileFaceNets trained with CosFace loss. (Left) a model trained on exclusively female images. (Right) a model trained on exclusively male images. 67
5.8 **Train Set Imbalance**. Results of experiments that change the train set gender presentation balance for MobileFaceNet and ResNet-152 models trained **without class-balanced sampling**. 68
5.9 **Test Set Imbalance.** Results of experiments that change the test set gender presentation balance for MobileFaceNet and ResNet-152 models trained **without class-balanced sampling**. 68
5.10 **Train & Test Set Imbalance.** Results of experiments that adjust the gender presentation balance in both the train and test set for MobileFaceNet and ResNet-152 models trained **without class-balanced sampling**. 69
5.11 **Train Set Imbalance.** Results of experiments testing models trained with different gender presentation balance on the InterRace dataset. These plots are analogous to the first row of Figures 5.2. 69

6.1 **[Brief Overview of Fairness in ML]** For the scope of this work, we consider only in-processing techniques and apply them to deep neural networks. We show that that overparametrized nature of neural networks is one reason why current techniques fail. 79
6.2 An illustration of gerrymandering; color denotes label, shape and outline are sensitive attributes. Model on the right is more fair to shape but less fair to outline. . 96
6.3 Gerrymandering behavior on CheXpert. The equal loss regularizer (in orange) achieves better parity along one demographic (sex, see Table 6.1 by making predictions more disparate along another demographic (age). For a model to be fair across age groups, the bars should all be of the same height. 96

Chapter 1: Introduction

Recently, deep learning has led to remarkable advancements across various domains enabling significant progress in fields such as computer vision, robotics, and natural language processing. Furthermore, today neural networks are omnipresent in industrial and research applications spanning various high stakes environments such as healthcare, self-driving cars, and finance. As deep learning models are increasingly integrated into these critical applications, it becomes imperative to examine their limitations and address potential challenges to ensure their safe and reliable use. In this work we address two key challenges of deep neural networks – their vulnerability to adversarial attacks and their tendency to produce biased predictions.

In adversarial attacks malicious actors deliberately manipulate input data to deceive the model and induce misclassifications or erroneous outputs. Adversarial attacks pose a serious threat to the reliability of deep learning systems, raising concerns regarding safety as well as potential privacy risks. In our work, we study adversarial robustness of deep learning models and develop defense mechanisms against poisoning attacks.

The second challenge is the issue of unfairness in deep learning models. As deep learning systems are increasingly deployed in industry, ensuring fairness becomes paramount since biases and discriminatory behavior within neural networks can lead to significant societal consequences. Developing an understanding of these biases is important for helping protect against

disproportionate harmful outcomes impacting vulnerable groups and perpetuating existing societal inequalities.

This thesis comprises a comprehensive exploration of these challenges in four parts and is organised as follows. In Chapter 3, we examine adversarial robustness of face recognition systems. Leveraging adversarial attacks, we develop a tool, LowKey, for modifying face images to protect social media users from unauthorized surveillance. Our system pre-processes user images before they are made publicly available on social media outlets so they cannot be used by a third party for facial recognition purposes. In particular LowKey works by moving the feature space representations of gallery faces so that they do not match corresponding probe images while preserving image quality. We find that our method effectively degrades the performance of even commercial-grade black-box face recognition APIs, whose inner workings are not publicly known. We released a research prototype of LowKey to the public through a web interface.

In Chapter 4, we explore defences against data poisoning and backdoor attacks which, in contrast to inference-time attacks, manipulate victim models by maliciously modifying training data. Many previous defenses against poisoning either fail in the face of increasingly strong attacks, or lead to significant accuracy trade-offs. We find that strong data augmentations, such as MixUp and CutMix, can significantly diminish the threat of poisoning and backdoor attacks while improving the model's performance.

After that we delve into studying the origins of unfairness in deep learning models. One major source of model bias is dataset imbalance with respect to a protected attribute. In Chapter 5 we focus on unraveling the complex effects that dataset imbalance can have on model bias for face identification systems. Interestingly, we find that the gallery set imbalance is as important as train set data imbalance and the effects of imbalance in train and gallery sets can amplify (in case

of images) or cancel (in case of identities) each other. We also observe that train and test class imbalances are not the only drivers of bias in face recognition systems.

Much effort has been devoted to understanding and correcting biases in classical machine learning models, where overfitting is not a pernicious issue and where fairness constraints imposed at train time often generalize to test data. In Chapter 6 we explore the behavior of existing methods for bias mitigation in neural networks, shedding light on their effectiveness and uncovering potential unintended consequences of algorithmic fairness approaches.

Chapter 2: Preliminaries

2.1 Neural Networks

According to the definition of a neural network from [Goodfellow et al., 2016],

The goal of a neural network is to approximate some function f^*. For example, for a classifier, $y = f^*(x)$ maps an input x to a category y. A neural network defines a mapping $y = f(x;\theta)$ and learns the value of the parameters θ that result in the best function approximation.

Neural networks can be represented as a composition of n functions, called layers

$$f(x) = f_n \circ f_{n-1} \circ ... \circ f_1(x),$$

which can be any linear or non-linear functions or a composition of functions. The simplest example of a neural network is a multi-layer perceptron, where each layer f_i is a linear function followed by a rectified linear activation function $f_i = \sigma(A_i x + b_i),\ \sigma(x) = [\max(x_i, 0)]$. The depth of a neural network refers to the number of layers in the model. Each layer f_i has a set of learnable parameters θ_i and the combination of parameters in all layers is denoted by θ. During neural network training, we optimize parameters of the model θ to drive $f(x)$ to match $f^*(x)$.

Convolutional Neural Networks (CNNs) are a specialized type of neural network architecture that are highly effective in computer vision tasks. Unlike traditional neural networks,

CNNs exploit the spatial structure of input data, making them well-suited for image analysis. The fundamental operation in a CNN is the convolutional layer, which consists of a set of learnable small-sized matrices, called filters, that are convolved with the input data to produce feature maps. By sliding the filters across the input, the CNN is able to capture local patterns and spatial dependencies. CNNs also incorporate pooling layers, which downsample the feature maps to reduce their spatial dimensions and make computation more efficient. In addition to convolutional and pooling layers, modern CNN architectures typically include residual connections, normalization layers, as well as fully connected layers at the end of the network.

One of the advantages of neural networks is that they can be used for automatic feature learning. During training, these systems learn representations of the input x at the intermediate layers $f_i(x)$, which can subsequently be used for the target task such as classification at the output layer.

Neural networks are trained using the principle of empirical risk minimization, which involves solving an optimization problem to minimize the expected loss of a model over a data distribution. The loss quantifies the discrepancy between the outputs of the network and the ground truth labels associated with inputs. More formally, consider a dataset of samples $D = \{(x_i, y_i)\}_{i=1}^{N}$ drawn from some distribution $\mathcal{D}$, then we solve the following optimization problem with respect to the network parameters:

$$\min_{\theta} \mathbb{E}_{\mathcal{D}} \mathcal{L}(f_\theta(x), y)$$

The choice of the loss function depends on the specific task. In this work we focus on image classification problems, where it is common to use cross-entropy loss function in conjunction

with softmax function. Let ϕ be the output vector of f_θ for some sample x with entries corresponding to classes in the dataset. Then, softmax function converts network output scores into a probability distruibution over the classes, and cross-entropy loss enforces the model to assign high score to the correct class:

$$s(\phi)_i = \frac{e^{\phi_i}}{\sum_j e^{\phi_j}}$$

$$c(\phi, y) = -\log(s(\phi)_y)$$

The minimization problem is typically solved with gradient-based methods, such as stochastic gradient descent.

2.2 Face Recognition

Throughout this work we extensively use face recognition systems in our experiments. In Chapter 3 we explore adversarial robustness of face recognition systems, in Chapter 5 we investigate the influence of data imbalance on bias in face identification, and in Chapter 6 we evaluate the effectiveness of bias mitigation techniques in the context of face recognition and medical image classification systems. In this section we detail necessary definitions and approaches in face recognition (FR).

State-of-the-art face recognition systems are based on neural networks. Those models are trained in a classification manner using special loss functions designed for better class separation. The main difference between a classification model and a facial recognition model is that the latter should be able to recognize images of people it has never seen during training, i.e. images from

new classes. To recognize photos of new identities (**probe images**), the system needs access to a database consisting of photos with known identities (**gallery images**). Then, to recognize a person on a probe image x_i, the model extracts its feature vector f_i and finds gallery images with the closest feature vectors in cosine distance using a k-nearest neighbors search. The matched images are used to reveal the identity.

Face Identification is the task of answering the question, "who is this person?" Identification entails comparing a probe image to gallery images in order to find potential matches.

Face Verification answers the question, "is this person who they say they are?", or equivalently "are these two photos of the same person?" Verification is used, for example, to unlock phones.

To enhance the discriminative power of the learned features, face recognition models are trained using specialized loss functions. These loss functions are designed to promote greater angular separation between features belonging to different classes, while simultaneously maintaining a compact intra-class distance. Two examples of such loss functions are CosFace [Wang et al., 2018] and ArcFace [Deng et al., 2019].

Let us re-formulate cross-entropy loss in terms in cosine similarity. For an input image x, the corresponding feature vector f and label y, the cross-entropy loss can be written as:

$$\mathcal{L}_{CE} = -\log \frac{e^{\phi_y}}{\sum_j e^{\phi_j}}, \ \phi_j = W_j^T f = \|W_j\| \|f\| \cos\theta_j$$

where W is the weight of the fully-connected layer, and θ_j is the angle between f and weight vector corresponding to $j-$th class W_j. Let us fix the norm of weight matrix of the fully-connected

layer W and feature vector f:

$$\|f\| = s, \;\; \|W\| = 1$$

Then, the normalized version of cross-entropy loss (NCE) can be written as

$$L_{NCE} = -\log \frac{e^{s\cos(\theta_y)}}{\sum_j e^{s\cos(\theta_j)}}$$

To promote better angular feature discrimination, CosFace loss introduces an additive angular margin to the traditional cross-entropy loss

$$\mathcal{L}_{\text{CosFace}} = -\log\left(\frac{e^{s(\cos(\theta_y)-m)}}{e^{s(\cos(\theta_y)-m)} + \sum_{j\neq y} e^{s\cos(\theta_j)}}\right)$$

subject to

$$W = \frac{W^*}{\|W^*\|}, \;\; f = \frac{f^*}{\|f^*\|}, \;\; \cos(\theta_j) = W_j^T f$$

Intuitively, adding angular margin penalizes the model for having a small cosine similarity between the feature vector of an image and the weight vector of the fully-connected layer corresponding to the correct label. Similarly, the ArcFace loss introduces an adaptive angular margin:

$$\mathcal{L}_{\text{ArcFace}} = -\log\left(\frac{e^{s(\cos(\theta_y+m)))}}{e^{s(\cos(\theta_y+m))} + \sum_{j\neq y} e^{s\cos(\theta_j)}}\right).$$

Chapter 3: LowKey: Leveraging Adversarial Attacks to Protect Social Media Users from Facial Recognition

Joint work with Micah Goldblum, Harrison Foley, Shiyuan Duan, John P Dickerson, Gavin Taylor and Tom Goldstein. Appeared at the International Conference on Learning Representations (ICLR), 2021.

Facial recognition systems are increasingly deployed by private corporations, government agencies, and contractors for consumer services and mass surveillance programs alike. These systems are typically built by scraping social media profiles for user images. Adversarial perturbations have been proposed for bypassing facial recognition systems. However, existing methods fail on full-scale systems and commercial APIs. We develop our own adversarial filter that accounts for the entire image processing pipeline and is demonstrably effective against industrial-grade pipelines that include face detection and large scale databases. Additionally, we release an easy-to-use webtool that significantly degrades the accuracy of Amazon Rekognition and the Microsoft Azure Face Recognition API, reducing the accuracy of each to below 1%.

Figure 3.1: Top: original images, Bottom: protected by LowKey.

3.1 Introduction

Facial recognition systems (FR) are widely deployed for mass surveillance by government agencies, government contractors, and private companies alike on massive databases of images belonging to private individuals [Hartzog, 2020, Derringer, 2019, Weise and Singer, 2020]. Recently, these systems have been thrust into the limelight in the midst of outrage over invasion into personal life and concerns regarding fairness [Singer, 2018, Lohr, 2018, Cherepanova et al., 2021]. Practitioners populate their databases by hoarding publicly available images from social media outlets, and so users are forced to choose between keeping their images outside of public view or taking their chances with mass surveillance.

We develop a tool, LowKey, for protecting users from unauthorized surveillance by leveraging methods from the adversarial attack literature, and make it available to the public as a webtool. LowKey is the first such evasion tool that is effective against commercial facial recognition APIs. Our system pre-processes user images before they are made publicly available on social media outlets so they cannot be used by a third party for facial recognition purposes. We establish the effectiveness of LowKey throughout this work.

Our contributions can be summarized as follows:

- We design a black-box adversarial attack on facial recognition models. Our algorithm moves the feature space representations of gallery faces so that they do not match corresponding probe images while preserving image quality.
- We interrogate the performance of our method on commercial black-box APIs, including Amazon Rekognition and Microsoft Azure Face, whose inner workings are not publicly known. We provide comprehensive comparisons with the existing data poisoning alternative, Fawkes [Shan et al., 2020], and we find that while Fawkes is ineffective in every experiment, our method consistently prevents facial recognition.
- We release an easy-to-use webtool, LowKey, so that social media users are no longer confronted with a choice between withdrawing their social media presence from public view and risking the repercussions of being surveilled.

3.2 Related Work

Neural networks are known to be vulnerable to *adversarial attacks*, small perturbations to inputs that do not change semantic content, and yet cause the network to misbehave [Goodfellow et al., 2014]. The adversarial attack literature has largely focused on developing new algorithms that, in simulations, are able to fool neural networks [Carlini and Wagner, 2017, Chiang et al., 2020]. Most works to date focus on the idea of *physical world attacks*, in which the attacker places adversarial patterns on an object in hopes that the adversarial properties transfer to an image of the object. Such attacks do not succeed reliably because the adversarial perturbation must survive imaging under various lighting conditions, object orientations, and occlusions [Kurakin

et al., 2018]. While researchers have succeeded in crafting such attacks against realistic systems, these attacks do not work consistently across environments [Wu et al., 2020, Xu et al., 2020, Goldblum et al., 2021]. In facial recognition, attacks have largely focused on physical backdoor threat models, evasion attacks on verification [Wenger et al., 2020, Zhong and Deng, 2020] and attacks on face detection [Pedraza et al., 2018]. Unlike these physical threat models, the setting in which we operate is purely *digital*, meaning that we can manipulate the contents of digital media at the bit level, and then hand manipulated data directly to a machine learning system. The ability to digitally manipulate media greatly simplifies the task of attacking a system, and has been shown to enhance transferability to black box industrial systems for applications like copyright detection [Saadatpanah et al., 2020] and financial time series analysis [Goldblum et al., 2021].

Recently, the Fawkes algorithm was developed for preventing social media images from being used by unauthorized facial recognition systems [Shan et al., 2020]. However, Fawkes, along with the experimental setup on which it is evaluated in the original work, suffers from critical problems. First, Fawkes assumes that facial recognition practitioners train their models on each individual's data. However, high-performance FR systems instead harness large pre-trained Siamese networks [Liu et al., 2017, Deng et al., 2019]. Second, the authors primarily use image *classifiers*. In contrast, commercial systems are trained with FR-specific heads and loss functions, as opposed to the standard cross-entropy loss used by classifiers. Third, the authors perform evaluations on very small datasets. Specifically, they test Fawkes against commercial APIs with a gallery containing only 50 images. Fourth, the system was only evaluated using top-1 accuracy, but FR users such as police departments often compile a list of suspects rather than a single individual. As a result, other metrics like top-50 accuracy are often used in facial recognition,

and are a more realistic metric for when a system has been successfully suppressed. Fifth, while the original work portrays Fawkes' perturbations are undetectable by the human eye, experience with the codebase suggests the opposite (indeed, a New York Times journalist likewise noted that the Fawkes images she was shown during a demonstration were visibly heavily distorted). Finally, Fawkes has not yet released an app or a webtool, and regular social media users are unlikely to make use of git repositories. Our attack avoids the aforementioned limitations, and we perform thorough evaluations on a large collection of images and identities. When comparing with Fawkes, we use the authors' own implementation in order to make sure that all evaluations are fair. Furthermore, we use Fawkes' highest protection setting to make sure that LowKey performs better than Fawkes' best attack. Another work uses targeted adversarial attack on probe images for facial recognition systems so that they cannot be matched with images in a database [Yang et al., 2021].

3.3 The LowKey Attack on Mass Surveillance

3.3.1 Problem Setup

To help make our work more widely accessible, we begin by introducing common facial recognition terms.

Gallery images are database images with known identities. These often originate from such sources as passport photos and social media profiles. The gallery is used as a reference for comparing new images.

Probe images are new photos whose subject the FR system user wants to identify. For example, probe images may be extracted from video surveillance footage. The extracted images

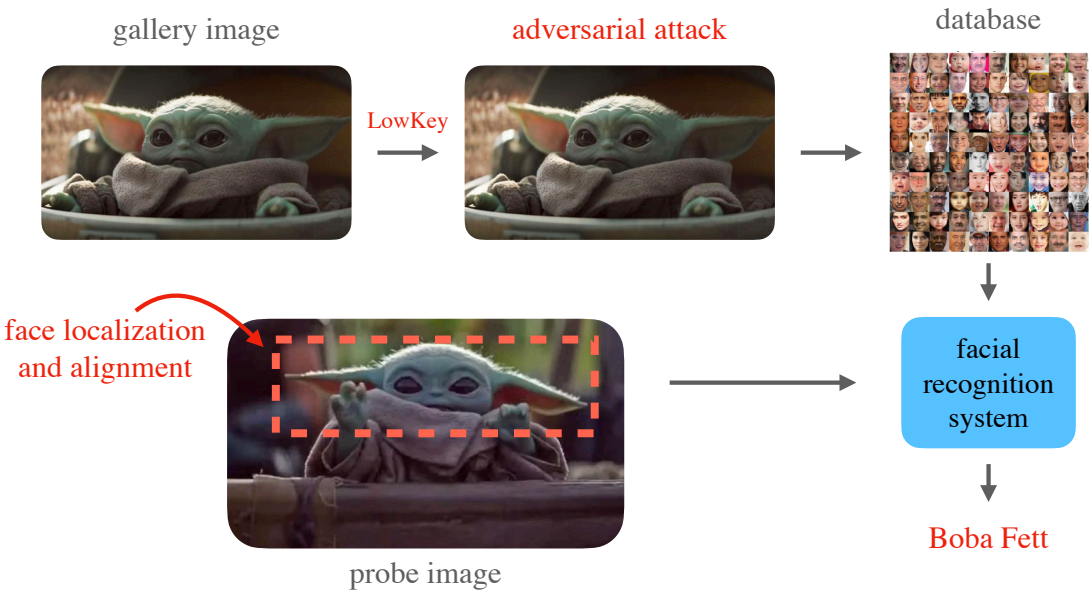

Figure 3.2: The LowKey pipeline. When users protect their publicly available images with LowKey, facial recognition systems cannot match these harvested images with new images of the user, for example from surveillance cameras.

are then fed into the FR system, and matches to gallery images with known identities.

Identification is the task of answering the question, "who is this person?" Identification entails comparing a probe image to gallery images in order to find potential matches. In contrast, **verification** answers the question, "is this person who they say they are?", or equivalently "are these two photos of the same person?" Verification is used, for example, to unlock phones.

In our work, we focus on identification, which can be used for mass surveillance. State-of-the-art facial recognition systems first detect and align faces before extracting facial features from the probe image using a neural network. These systems then find gallery images with the closest feature vectors using a k-nearest neighbors search. The matched gallery images are then considered as likely identities corresponding to the person in the probe photo. LowKey applies a filter to user images which may end up in an organization's database of gallery images. The result is to corrupt the gallery feature vectors so that they will not match feature vectors corresponding

to the user's probe images. A visual depiction of the LowKey pipeline can be found in Figure 3.2.

3.3.2 The LowKey attack

LowKey manipulates potential gallery images so that they do not match probe images of the same person. LowKey does this by generating a perturbed image whose feature vector lies far away from the original image, while simultaneously minimizing a perceptual similarity loss between the original and perturbed image. Maximizing the distance in feature space prevents the image from matching other images of the individual, while the perceptual similarity loss prevents the image quality from degrading. In this section, we formulate the optimization problem, and describe a number of important details.

LowKey is designed to evade proprietary FR systems that contain pre-processing steps and neural network backbones that are not publicly known. In order to improve the transferability of our attack to unknown facial recognition systems, LowKey simultaneously attacks an ensemble of models with various backbone architectures that are produced using different training algorithms. Additionally, for each model in the ensemble, the objective function considers the locations of feature vectors of the attacked image both with and without a Gaussian blur. We find that this technique improves both the appearance and transferability of attacked images. Experiments and ablations concerning ensembling and Gaussian smoothing can be found in Section 3.6. For perceptual similarity loss, we use LPIPS, a metric based on ℓ_2 distance in the feature space of an ImageNet-trained feature extractor [Zhang et al., 2018]. LPIPS has been used effectively in the image classification setting to improve the image quality of adversarial examples [Laidlaw et al.,

2021].

Formally, the optimization problem we solve is

$$\max_{x'} \frac{1}{2n} \sum_{i=1}^{n} \frac{\overbrace{\|f_i(A(x)) - f_i(A(x'))\|_2^2}^{\text{non-smoothed}} + \overbrace{\|f_i(A(x)) - f_i(A(G(x')))\|_2^2}^{\text{smoothed}}}{\|f_i(A(x))\|_2} - \alpha \underbrace{\text{LPIPS}(x, x')}_{\text{perceptual loss}}, \quad (3.1)$$

where x is the original image, x' is the perturbed image, f_i denotes the i^{th} model in our ensemble, G is the Gaussian smoothing function with fixed parameters, and A denotes face detection and extraction followed by 112×112 resizing and alignment. The face detection step is an important part of the LowKey objective function, as commercial systems rely on face detection and extraction because probe images often contain a scene much larger than a face, or else contain a face who's alignment is not compatible with the face recognition system.

We solve this maximization problem iteratively with signed gradient ascent, which is known to be highly effective for breaking common image classification systems [Madry et al., 2017a]. Namely, we iteratively update x' by adding the sign of the gradient of the maximization objective (3.1) with respect to x'. By doing this, we move x' and $G(x')$ far away from the original image x in the feature spaces of models f_i used in the LowKey ensemble. The ensemble contains four feature extractors, IR-152 and ResNet-152 backbones trained with ArcFace and CosFace heads. More details can be found in the next section.

Additional details concerning attack hyperparameters can be found in Section 3.8.1.

3.4 Experimental Design

Our ensemble of models contains ArcFace and CosFace facial recognition systems [Deng et al., 2019, Wang et al., 2018]. For each of these systems, we train ResNet-50, ResNet-152, IR-50, and IR-152 backbones on the MS-Celeb-1M dataset, which contains over five million images from over 85,000 identities [He et al., 2016, Deng et al., 2019, Guo et al., 2016]. We use these models both in our ensemble to generate attacks and to perform controlled experiments in Section 3.6. Additional details on our models and their training routines can be found in Section 3.8.1.

We primarily test our attacks on the FaceScrub dataset, a standard identification benchmark from the MegaFace challenge, which contains over 100,000 images from 530 known identities as well as one million distractor images [Kemelmacher-Shlizerman et al., 2016]. We discard near-duplicate images from the dataset as is common practice in the facial recognition literature [Zhang et al., 2020]. We also perform experiments on the UMDFaces dataset, which can be found in Section 3.8.3 [Bansal et al., 2017]. We treat one tenth of each identity's images as probe images, and we insert the remaining images into the gallery. We randomly select 100 identities and apply LowKey to each of their gallery images. This setting simulates a small pool of LowKey users among a larger population of non-users. Then, in order to perform a single evaluation trial of identification, we randomly sample one probe image from a known identity and find its closest matches within the remainder of the FaceScrub dataset, according to the facial recognition model. Distance is measured in feature space of the model. If the FR model selects a match from the same identity, then the trial is a success.

Rank-k Accuracy. For each probe image, we consider the model successful in the rank-k setting if the correct identity appears among the k closest gallery images in the model's feature

	Amazon rank-1	Amazon rank-50	Microsoft rank-1
Clean	93.7%	95.4%	90.5%
Fawkes	77.5%	94.9%	74.2%
LowKey	0.6%	2.4%	0.1%

Table 3.1: An evaluation of Amazon Rekognition and Microsoft Azure Face on FaceScrub data with LowKey and Fawkes protection (a small number, and lighter color, indicates a successful attack). LowKey consistently achieves virtually flawless protection, while Fawkes provides little protection.

space. To test the transferability of our attack we compute rank-1 and rank-50 accuracy for attack, and test feature extractors from our set of trained FR models.

3.5 Breaking Commercial Black-Box APIs

The ultimate test for our protection tool is against commercial systems. These systems are proprietary, and their exact specifications are not publicly available. We test LowKey in the black-box setting using two commercial facial recognition APIs: Amazon Rekognition and Microsoft Azure Face. We also compare against Fawkes. We generate Fawkes images using the authors' own code and hyperparameters to ensure a fair comparison, and we use the highest protection setting their code offers.

Amazon Rekognition Amazon Rekognition is a commercial tool for detecting and recognizing faces in photos. Rekognition works by matching probe images with uploaded gallery images that have known labels. Amazon does not describe how their algorithm works, but their approach seemingly does not involve training a model on uploaded images (at least not in a supervised manner). We test the Rekognition API using the FaceScrub dataset (including distractors) where 100 randomly selected identities have their images attacked as described in Section 3.4.

We observe that LowKey is highly effective, and even in the setting of rank-50 accuracy, Rekognition can only recognize 2.4% of probe images belonging to users protected with LowKey. In contrast, Fawkes fails, with 77.5% of probe images belonging to its users recognized correctly in the rank-1 setting and 94.9% of these images recognized correctly when the 50 closest matches are considered. This is close to the performance of Amazon Rekognition on clean images.

Microsoft Azure Face We repeat a similar experiment on the Microsoft Azure Facial Recognition API. In contrast to Amazon's API, Microsoft updates their model on the uploaded gallery of images. Therefore, only known identities can be used, so we only include images corresponding to the 530 known identities from FaceScrub and no distractors. The Azure system recognizes only 0.1% of probe images whose gallery images are under the protection of LowKey. Even though Fawkes is designed to perform data poisoning, and authors claim it is especially well suited to Microsoft Azure Face, in our experiments, Azure is still able to recognize more than 74% of probe images uploaded by users who employ Fawkes.

We conclude from these experiments that LowKey is both highly effective and transferable to even state-of-the-art industrial facial recognition systems. In the next section, we explore several components of our attack in order to uncover the tools of its success.

3.6 Additional Experiments

The effectiveness of our protection tool hinges on several properties:

1. The attack must transfer effectively to unseen models.

2. Images must look acceptable to users.

3. LowKey must run sufficiently fast so that run-time does not outweigh its protective benefits.

4. Attacked images must remain effective after being saved in PNG and JPG formats.

5. The algorithm must scale to images of any size.

We conduct extensive experiments in this section with a variety of facial recognition systems to interrogate these properties of LowKey.

3.6.1 Ensembles and Transferability

In developing the ensemble of models used to compute our attack, we examine the extent to which attacks generated by one model are effective against another. By including an eclectic mix of models in our ensemble, we are able to ensure that LowKey produces images that fool a wide variety of facial recognition systems. To this end, we evaluate attacks on all pairs of source and victim models with ResNet-50, ResNet-152, IR-50, and IR-152 backbones, and both ArcFace and CosFace heads. For each victim model, we additionally measure performance on clean images, our ensembled attack, and Fawkes. See Table 3.2 for a comparison of the rank-50 performance of these combinations. Additional evaluations in the rank-1 setting and on the UMDFaces dataset can be found in Section 3.8.2 and 3.8.3 respectively. Note that entries for which the attacker and defender models are identical depict white-box performance, while entries for which these model differ depict black-box transferability.

We observe in these experiments that adversarial attacks generated by IR architectures transfer better to IR-based facial recognition systems, while attacks generated by ResNet architectures transfer better to other ResNet systems. In general, attacks computed on 152-layer backbones are more effective than attacks computed on 50-layer backbones, and deeper networks are also more difficult to fool. Moreover, attacks transfer better between models trained with the

Attacker \ Defender	IR-50A	IR-50C	IR-152A	IR-152C	RN-50A	RN-50C	RN-152A	RN-152C
Clean	96.8%	96.8%	96.7%	96.8%	96.8%	96.8%	96.7%	96.7%
Fawkes	96.6%	96.7%	96.7%	96.7%	96.7%	96.5%	96.6%	96.6%
IR-50A	0.4%	22.2%	11.9%	35.2%	33.6%	46.4%	45.7%	53.0%
IR-50C	4.9%	0.3%	4.1%	8.0%	23.1%	25.9%	31.4%	28.6%
IR-152A	9.9%	18.3%	0.1%	26.1%	41.6%	46.2%	49.6%	48.9%
IR-152C	2.8%	1.6%	1.5%	0.5%	11.9%	13.9%	18.8%	16.3%
RN-50A	26.4%	35.7%	36.3%	43.0%	0.9%	13.3%	17.4%	24.2%
RN-50C	33.8%	36.5%	41.1%	42.9%	9.9%	0.2%	17.9%	21.1%
RN-152A	16.8%	22.0%	21.1%	28.2%	5.2%	8.8%	0.3%	7.6%
RN-152C	14.8%	19.2%	19.9%	24.3%	6.7%	6.9%	7.1%	0.5%
Ensemble	3.0%	2.4%	2.1%	0.6%	3.1%	4.2%	5.5%	0.9%

Table 3.2: Rank-50 accuracy of the LowKey and Fawkes attacks. After the first two rows, each row represents LowKey attacks generated from the same model. Each column represents inference on a single model. The first two letters in the model's name denote the type of backbone: IR or ResNet (RN). The last letter in the model's name indicates the type of head; "A" denotes ArcFace, and "C" denotes CosFace. Smaller numbers, and lighter colors, indicate more successful attacks.

same head. An ensemble of models of all combinations of ResNet-152 and IR-152 backbones as well as ArcFace and CosFace heads generates attacks that transfer effectively to all models and fool models at only a slightly lower rate than white-box attacks.

3.6.2 Gaussian Smoothing

We incorporate Gaussian smoothing as a pre-processing step in our objective function (3.1) to make our perturbations smoother and more robust. Intuitively, this promotes the effectiveness of the attacked image even when a denoising filter is applied. The presence of blur forces the adversarial perturbation to rely on smoother/low-frequency image modifications rather than adversarial "noise." Empirically, we find that attacks computed with this procedure produce slightly

Attacker \ Defender	IR-50A	IR-50C	IR-152A	IR-152C	RN-50A	RN-50C	RN-152A	RN-152C
Clean	96.7%	96.8%	96.6%	96.9%	96.7%	96.7%	96.7%	96.7%
Without GS	78.6%	74.0%	74.4%	64.6%	75.5%	76.0%	77.8%	74.2%
With GS	4.2%	4.8%	4.4%	2.8%	7.9%	7.1%	9.6%	3.2%

Table 3.3: Rank-50 accuracy of FR models tested on blurred LowKey images computed with/without Gaussian smoothing.

smoother and more aesthetically pleasing perturbations without sharp lines and high-frequency oscillations. See Figure 3.3 for a visual comparison of images produced with and without Gaussian smoothing in the LowKey pipeline.

We additionally produce images both with and without smoothing in the attack pipeline. Before feeding them into facial recognition systems, we defend the system against our attacks by applying a Gaussian smoothing pre-processing step just before inference.

We find that facial recognition systems which use this pre-processing step perform equally well on rank-50 (but not rank-1) accuracy compared to performance without smoothing, and they are also able to defeat attacks which are not computed with Gaussian smoothing. On the other hand, attacks computed using Gaussian smoothing are able to counteract this defense and fool the facial recognition system (see Table 3.3). This suggests that attacks that use Gaussian smoothing in their pipeline are more robust and harder to defend against. See Section 3.8.6 for details regarding Gaussian smoothing hyperparameters.

3.6.3 Run-Time

In order for users to be willing to use our tool, LowKey must run fast enough that it is not an inconvenience to use. Computing adversarial attacks is a computationally expensive task. We

Figure 3.3: LowKey attacked images computed without (above) and with (below) Gaussian smoothing.

compare run-time to Fawkes as a baseline and test both attacks on a single NVIDIA GeForce RTX 2080 TI GPU. We attack one image at a time with no batching for fair comparison, and we average over runs on every full-size gallery image from each of five randomly selected identities from FaceScrub. While Fawkes averages 54 seconds per image, LowKey only averages 32 seconds per image. In addition to providing far superior protection, LowKey runs significantly faster than the existing method, providing users a smoother and more convenient experience.

3.6.4 Robustness to Image Compression

Since users may save their images in various formats after passing them through LowKey, the images we produce must provide protection even after being saved in common formats. Our baseline tests are conducted with images saved in uncompressed PNG format. To test performance under compression, we convert protected images to JPEG format and repeat our experiments on commercial APIs. While compression very slightly decreases performance, the attack

Figure 3.4: First row: Original large images, Second row: Images protected with LowKey (medium magnitude), Third row: Images protected with LowKey (large magnitude).

is still very effective: Microsoft Azure Face is now able to recognize 0.2% of images compared to 0.1% when saved in the PNG format. Likewise, Amazon Rekognition now recognizes 3.8% of probe images compared to 2.4% previously.

3.6.5 Scalability to All Image Sizes (Disclaimer)

Many tools in deep learning require that inputs be of particular dimensions, but user images on social media sites come in all shapes and sizes. Therefore, LowKey must be flexible. Since the detection and alignment pipeline in our attack resizes images in a differentiable fashion, we can

attack images of any size and aspect ratio. Additionally, we apply the LPIPS penalty to the entire original image, which prevents box-shaped artifacts from developing on the boundaries of the rectangle containing the face. Since LowKey does not have a fixed attack budget, perturbations may have different magnitudes on different images. Figure 3.4 shows the variability of LowKey perturbations on very large images; the image of Tom Hanks (first column) is one of the best looking examples of LowKey on large images, while the image of Tina Fey (last column) is one of the worst looking examples. Protecting very large images is a more challenging task than protecting small images because of the black-box detection, alignment, and re-scaling used in APIs which affect large images more significantly. These experiments indicate that users will receive stronger protection if they use LowKey on smaller images.

We test the effectiveness of LowKey on large images by protecting gallery images of 10 identities from Facescrub (with 17 images in the gallery on average) and using 20 probe images per person. We also vary the magnitude of the perturbation to find the smallest perturbation that is sufficient to protect images (Table 3.4). In this way, we find that users may trade off some protection in exchange for better looking images at their own discretion. Additionally, we find that LowKey works much better with smaller gallery sizes; when only 5 gallery images are used, the performance of Amazon Rekognition drops from 32.5% to 11% in the rank-50 setting. This observation suggests that users can upload new profile pictures less frequently in order to decrease the number of gallery images corresponding to their identity and thus enhance their protection. Finally, the quality of probe images is also important; when small probe images are used, like those which would occur in low resolution security camera footage, the accuracy of Amazon Rekognition drops from 32.5% to 19%.

	Amazon rank-1	Amazon rank-50	Microsoft rank-1
Clean	89.0%	98.5%	86.0%
LowKey 10	63.0%	94.5%	75.5%
LowKey 20	34.0%	59.5%	30.5%
LowKey 30	20.5%	36.5%	12.7%
LowKey 40	14.5%	36.0%	3.0%
LowKey 50	11.0%	32.5%	0.0%

Table 3.4: Evaluation of LowKey on full-size images. Rows indicate levels of magnitude of LowKey (denoted by the number of attack steps).

3.7 Discussion

In this work, we develop a tool for protecting users from unauthorized facial recognition. Our tool adversarially pre-processes user images before they are uploaded to social media. These pre-processed images are useless for third-party organizations who collect them for facial recognition. While we have shown that LowKey is highly effective against commercial black-box APIs, it does not protect users 100% of the time and may be circumvented by specially engineered robust systems. Thus, we hope that users will still remain cautious about publicly revealing personal information. One interesting future direction is to produce adversarial filters that are more aesthetically pleasing in order to promote wider use of this tool. However, it may be that there is no free lunch, and one cannot fool state-of-the-art facial recognition systems without visible perturbations. Facial recognition systems are not fragile, and other attacks that have attempted to break them have failed. Finally, we note that one of our goals in making this tool widely available is to promote broader awareness of facial recognition and the ethical issues it raises. Our webtool can be found at `lowkey.umiacs.umd.edu`.

3.8 Experimental Details and Additional Results

3.8.1 Implementation Details

We train all of our feature extractors using focal loss [Lin et al., 2017] with a batch size of 512 for 120 epochs. We use an initial learning rate of 0.1 and decrease it by a factor of 10 at epochs 35, 65 and 95. For the optimizer, we use SGD with a momentum of 0.9 and weight decay of 5e-4.

For our adversarial attacks, we use 0.05 for the perceptual similarity penalty, $\sigma = 3$ and window size 7 for the Gaussian smoothing term. Attacks are computed using signed SGD for 50 epochs with a learning rate of 0.0025.

For face detection and aligning models as well as for training routines, we use the face.evoLVe.PyTorch github repository [Zhao, 2020].

3.8.2 Rank-1 accuracy on FaceScrub data

See Table 3.5.

3.8.3 Results on UMDFaces dataset

We repeat controlled experiments on the UMDFaces dataset which contains over 367,000 photos of 8,277 identities. For UMDFaces, we also choose 100 identities at random and attack their gallery images while keeping one-tenth of each identity's photos as probe images. Experimental results are reported in Tables 3.6 and 3.7. It can be seen that the effectiveness of LowKey attacks on the UMDFaces dataset is slightly lower, which is likely a result of the much smaller

Attacker \ Defender	IR-50A	IR-50C	IR-152A	IR-152C	RN-50A	RN-50C	RN-152A	RN-152C
Clean	95.6%	96.1%	96.0%	96.2%	95.8%	95.9%	95.9%	96.0%
Fawkes	71.2%	76.2%	74.4%	78.2%	73.9%	76.0%	76.0%	71.2%
IR-50A	0.0%	3.5%	0.5%	6.4%	5.0%	8.2%	8.7%	12.4%
IR-50C	0.1%	0.0%	0.1%	9.0%	3.3%	3.3%	3.6%	5.3%
IR-152A	0.3%	1.8%	0.0%	4.2%	6.8%	8.9%	9.4%	10.8%
IR-152C	0.1%	0.1%	0.1%	0.0%	1.0%	1.4%	2.2%	2.9%
RN-50A	3.8%	7.0%	6.8%	8.3%	0.9%	2.0%	2.8%	4.6%
RN-50C	3.1%	5.7%	5.6%	7.9%	1.2%	0.1%	2.0%	3.5%
RN-152A	1.5%	3.1%	2.7%	4.9%	0.3%	0.5%	0.0%	0.4%
RN-152C	1.7%	2.6%	3.1%	3.9%	0.8%	0.8%	0.5%	0.0%
Ensemble	0.0%	0.0%	0.1%	0.0%	0.2%	0.4%	0.6%	0.1%

Table 3.5: Rank-1 accuracy of the LowKey and Fawkes attacks on the FaceScrub dataset. After the first two rows, each row represents LowKey attacks generated from the same model. Each column represents inference on a single model. The first two letters in the model's name denote the type of backbone: IR or ResNet (RN). The last letter in the model's name indicates the type of head; "A" denotes ArcFace, and "C" denotes CosFace. Smaller numbers, and lighter colors, indicate more successful attacks.

gallery.

3.8.4 Can we reduce the size of our attack?

In order to make our attacks more aesthetically pleasing, we try to reduce the size of perturbation by increasing the perceptual similarity penalty from 0.05 to 0.08. This attack is depicted in Figure 3.5 as a "LowKey small attack". Unfortunately, even a small decrease in the perturbation size results in a huge decrease in efficiency of the attack. In the rank-50 setting Amazon Rekognition is able to recognize 17.2% of probe images belonging to users protected with a LowKey small attack. Similarly, Microsoft Azure Face recognizes 5.5% of probe images. Results of controlled experiments are reported in Tables 3.8 and 3.9.

Attacker \ Defender	IR-50A	IR-50C	IR-152A	IR-152C	RN-50A	RN-50C	RN-152A	RN-152C
Clean	98.6%	98.3%	98.6%	98.6%	98.3%	98.3%	98.6%	98.6%
IR-50A	0.3%	38.9%	25.9%	48.6%	47.7%	55.7%	57.1%	62.2%
IR-50C	13.6%	0.3%	16.5%	17.6%	37.5%	39.2%	51.4%	46.3%
IR-152A	23.0%	32.1%	0.0%	36.4%	52.8%	56.5%	58.2%	58.5%
IR-152C	8.8%	8.8%	9.1%	1.1%	26.7%	28.1%	34.9%	31.8%
RN-50A	51.4%	56.3%	47.2%	53.1%	0.3%	30.7%	38.4%	43.2%
RN-50C	49.4%	48.3%	55.1%	54.0%	23.0%	1.1%	37.2%	36.4%
RN-152A	30.4%	38.1%	39.8%	43.8%	18.8%	22.2%	3.4%	25.9%
RN-152C	26.4%	33.8%	35.5%	37.8%	17.3%	18.2%	16.8%	3.4%
Ensemble	10.5%	4.5%	9.7%	8.8%	15.1%	6.5%	12.8%	13.6%

Table 3.6: Rank-50 accuracy of LowKey attacks on the UMDFaces dataset. After the first row, each row represents LowKey attacks generated from the same model. Each column represents inference on a single model. The first two letters in the model's name denote the type of backbone: IR or ResNet (RN). The last letter in the model's name indicates the type of head; "A" denotes ArcFace, and "C" denotes CosFace. Smaller numbers, and lighter colors, indicate more successful attacks.

3.8.5 Comparison with Fawkes

By comparing a set of images protected with LowKey and Fawkes tools, we can see that both attacks are noticeable, but distort images in different ways. While Fawkes adds conspicuous artifacts on the face (such as mustaches or lines on the nose), LowKey attack mostly changes the textures and adds spots on a person's skin. See Figure 3.5 for a visual comparison.

3.8.6 Gaussian Smoothing in LowKey

For the parameters of the Gaussian smoothing term in the optimization problem (3.1), we use 3 for σ and 7 for window size. For the defensive Gaussian blur, we use $\sigma = 2$ and no window size.

Attacker \ Defender	IR-50A	IR-50C	IR-152A	IR-152C	RN-50A	RN-50C	RN-152A	RN-152C
Clean	96.0%	97.2%	96.9%	96.9%	96.3%	96.6%	97.2%	96.3%
IR-50A	0.0%	12.2%	6.3%	18.2%	17.6%	25.9%	28.1%	31.3%
IR-50C	1.7%	0.0%	2.6%	4.3%	11.6%	13.9%	17.3%	21.0%
IR-152A	4.3%	12.8%	0.0%	14.5%	22.4%	25.6%	29.5%	30.1%
IR-152C	1.7%	2.0%	2.6%	0.0%	6.8%	9.7%	13.6%	11.4%
RN-50A	26.4%	31.0%	13.9%	26.4%	0.0%	9.4%	15.1%	19.0%
RN-50C	14.5%	21.3%	20.7%	25.3%	6.5%	0.6%	12.8%	14.8%
RN-152A	9.7%	16.2%	16.8%	17.9%	5.1%	7.1%	0.3%	7.4%
RN-152C	7.4%	10.2%	11.9%	13.1%	3.1%	4.5%	5.4%	0.9%
Ensemble	2.8%	1.7%	2.6%	3.1%	5.1%	2.0%	3.1%	4.0%

Table 3.7: Rank-1 accuracy of LowKey attack attacks on the UMDFaces dataset. After the first row, each row represents LowKey attacks generated from the same model. Each column represents inference on a single model. The first two letters in the model's name denote the type of backbone: IR or ResNet (RN). The last letter in the model's name indicates the type of head; "A" denotes ArcFace, and "C" denotes CosFace. Smaller numbers, and lighter colors, indicate more successful attacks.

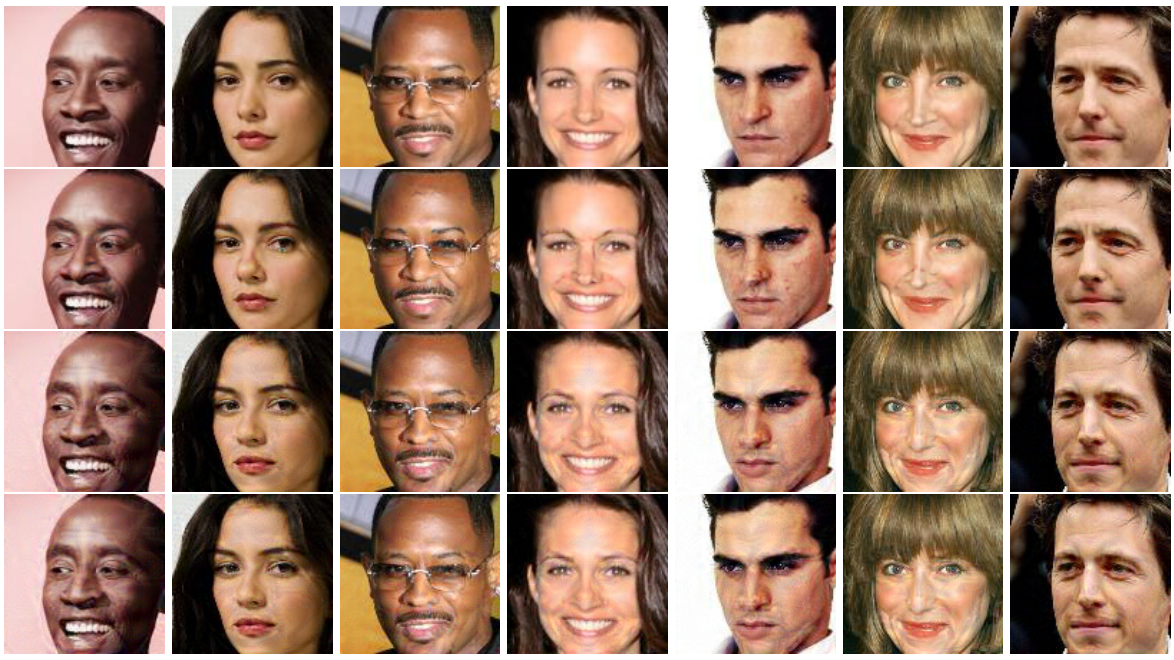

Figure 3.5: Panel of different attacks. First row: original images, second row: Fawkes attack, third row: LowKey small attack, last row: LowKey attack.

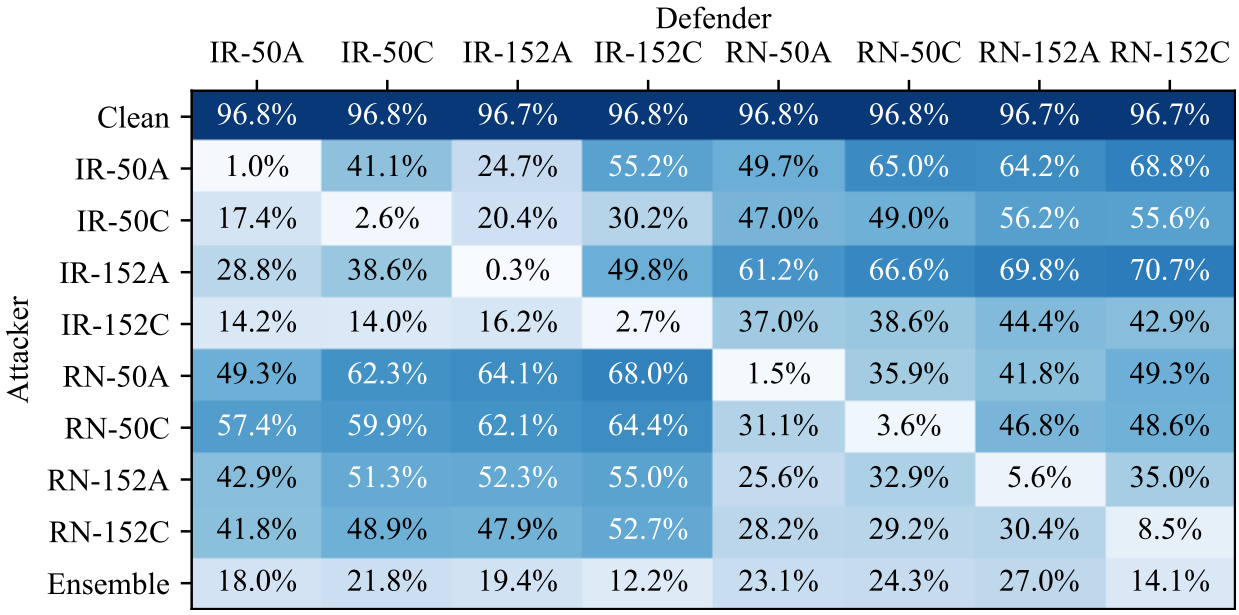

Table 3.8: Rank-50 accuracy of LowKey small attacks on the UMDFaces dataset. After the first row, each row represents LowKey small attacks generated from the same model. Each column represents inference on a single model. The first two letters in the model's name denote the type of backbone: IR or ResNet (RN). The last letter in the model's name indicates the type of head; "A" denotes ArcFace, and "C" denotes CosFace. Smaller numbers, and lighter colors, indicate more successful attacks.

Attacker \ Defender	IR-50A	IR-50C	IR-152A	IR-152C	RN-50A	RN-50C	RN-152A	RN-152C
Clean	95.6%	96.1%	96.0%	96.2%	95.8%	95.9%	95.9%	96.0%
IR-50A	0.0%	6.1%	2.2%	10.5%	9.7%	14.4%	16.9%	19.5%
IR-50C	1.6%	0.2%	2.5%	5.8%	9.3%	11.1%	14.0%	14.5%
IR-152A	2.2%	5.6%	0.0%	9.6%	14.0%	17.4%	19.2%	20.7%
IR-152C	1.3%	1.7%	2.2%	0.4%	5.1%	7.5%	9.6%	9.7%
RN-50A	8.7%	14.9%	14.8%	17.1%	0.2%	7.6%	7.8%	12.5%
RN-50C	9.9%	14.6%	15.6%	17.2%	5.6%	0.6%	8.8%	11.6%
RN-152A	9.2%	13.5%	12.2%	14.8%	5.7%	7.9%	0.8%	8.5%
RN-152C	6.9%	11.3%	11.8%	12.2%	4.8%	5.3%	6.0%	1.3%
Ensemble	2.6%	3.6%	3.5%	2.9%	4.0%	4.7%	5.4%	3.9%

Table 3.9: Rank-1 accuracy of LowKey small attacks on the UMDFaces dataset. After the first row, each row represents LowKey small attacks generated from the same model. Each column represents inference on a single model. The first two letters in the model's name denote the type of backbone: IR or ResNet (RN). The last letter in the model's name indicates the type of head; "A" denotes ArcFace, and "C" denotes CosFace. Smaller numbers, and lighter colors, indicate more successful attacks.

Attacker \ Defender	IR-50A	IR-50C	IR-152A	IR-152C	RN-50A	RN-50C	RN-152A	RN-152C
Clean	84.4%	85.3%	84.7%	86.1%	86.7%	87.9%	88.5%	89.6%
Without GS	13.3%	17.9%	15.8%	13.7%	18.9%	20.4%	23.2%	20.6%
With GS	0.3%	0.3%	0.1%	0.0%	1.0%	0.9%	0.6%	0.6%

Table 3.10: Rank-1 accuracy of FR models tested on blurred images attacked without and with the Gaussian smoothing term. The first two letters in the model's name denote the type of backbone: IR or ResNet (RN). The last letter in the model's name indicates the type of head; "A" denotes ArcFace, and "C" denotes CosFace. Smaller numbers, and lighter colors, indicate more successful attacks.

Chapter 4: Strong Data Augmentation Sanitizes Poisoning and Backdoor Attacks Without an Accuracy Tradeoff

Joint work with Eitan Borgnia, Liam Fowl, Amin Ghiasi, Jonas Geiping, Micah Goldblum, Tom Goldstein and Arjun Gupta. Appeared at the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2021.

Data poisoning and backdoor attacks manipulate victim models by maliciously modifying training data. In light of this growing threat, a recent survey of industry professionals revealed heightened fear in the private sector regarding data poisoning. Many previous defenses against poisoning either fail in the face of increasingly strong attacks, or they significantly degrade performance. However, we find that strong data augmentations, such as mixup and CutMix, can significantly diminish the threat of poisoning and backdoor attacks without trading off performance. We further verify the effectiveness of this simple defense against adaptive poisoning methods, and we compare to baselines including the popular differentially private SGD (DP-SGD) defense. In the context of backdoors, CutMix greatly mitigates the attack while simultaneously increasing validation accuracy by 9%.

4.1 Introduction

Machine learning models have demonstrated tremendous success in many domains from mobile image processing to security services [Schwartz et al., 2019, Lovisotto et al., 2020]. The growing availability of vast datasets has aided in this recent success. Practitioners often rely upon data scraped from the web or sourced from a third party [Papernot, 2018], where the security of such data can be compromised by malicious actors. *Data Poisoning* attacks pose a specific threat in which an attacker modifies a victim's training data to achieve goals such as targeted misclassification or performance degradation [Goldblum et al., 2022]. Basic data poisoning schemes implement backdoor triggers in training data, whereas recent works have also demonstrated that data poisoning schemes can successfully attack deep learning models trained on industrial-scale datasets without perceptible modifications [Huang et al., 2020, Geiping et al., 2021]. The seriousness of these threats is acknowledged by industrial practitioners, who recently ranked poisoning as the most worrisome threat to their interests [Kumar et al., 2020]. Furthermore, defenses designed for older, less powerful poisoning strategies work by filtering out poisons based on feature anomalies [Rubinstein et al., 2009] but fail when models are trained from scratch on poisoned data [Peri et al., 2019, Geiping et al., 2021]. Currently, the only method to prevent state-of-the-art targeted poisoning relies upon differentially private SGD (DP-SGD) and leads to a significant drop in validation accuracy [Geiping et al., 2021, Hong et al., 2020].

On the other hand, data augmentation has been a boon to practitioners, aiding in state-of-the-art performance on a variety of tasks [Zhang et al., 2017a, Gong et al., 2020]. Data augmentation can be used in many regimes, including settings where data is sparse, to improve generalization [Ni et al., 2021]. Simple augmentations include random crops or horizontal flips.

Recently, more sophisticated augmentation schemes have emerged that improve model performance: *mixup* takes pairwise convex combinations of randomly sampled training data and uses the corresponding convex combinations of labels. Not only does this prevent memorization of corrupt labels and provide robustness to adversarial examples, but it has also been shown to improve generalization [Zhang et al., 2017a]. Another augmentation technique, cutout, randomly erases patches of training data [DeVries and Taylor, 2017], whereas CutMix instead combines pairwise randomly sampled training data by taking random patches from one image and overlaying these patches onto other images [Yun et al., 2019]. The labels are then mixed proportionally to the area of these patches. CutMix enhances model robustness, achieves better test accuracy, and improves localization ability by encouraging the network to correctly classify images from a partial view. Finally, MaxUp applies a set of data augmentation techniques (basic or complex) to the training data and chooses the augmentation method and parameters that achieve the worst model performance of all the techniques [Gong et al., 2020]. By training against the most “difficult" data augmentation, MaxUp is able to improve generalization and, in some cases, adversarial robustness.

We investigate the effects of multiple augmentation strategies on data poisoning attacks. We find that these modern data augmentation strategies are often more effective than previous more cumbersome defenses against poisoning while also not sacrificing significant natural validation accuracy. Our data augmentation defense is additionally convenient for practitioners as it involves only a small and easy-to-implement change to standard training pipelines.

We empirically analyze this defense both in the setting of a simple backdoor trigger attack and in the setting of a modern imperceptible targeted data poisoning attack [Geiping et al., 2021].

4.2 Threat Model

There are many different flavors of poisoning attacks. In this work, we focus on two attacks from both sides of the spectrum: A simple and robust backdoor trigger attack and a modern poisoning attack that is *targeted* and also *clean-label*. Backdoor trigger attacks insert a specific trigger pattern (usually a small patch, but sometimes an additive perturbation or non-rectangular symbol) into training data. If this pattern is then added to images at test time, the network will misclassify the test image, assigning the label that was placed on the training images poisoned at train time. In contrast, *targeted* attacks are those in which the attacker wishes to modify the victim model to specifically misclassify a set of target images at inference time. Such attacks are often *clean-label*, meaning the modified training images retain their semantic content and are labeled correctly. Such attacks are optimization-based, finding the most effective perturbation of training data using gradient descent. This can make the attack especially hard to detect for sanitization-based defenses because it does not significantly degrade clean accuracy [Huang et al., 2020, Geiping et al., 2021]. We focus on both backdoor and clean-label attacks because of their renewed and recent interest in the community and because they cover the poisoning literature from two sides.

Any poisoning threat model can be formally described as a bilevel problem. Let $F(x, \theta)$ be a neural network taking inputs $x \in \mathbb{R}^n$ with parameters $\theta \in \mathbb{R}^p$. The attacker is allowed to modify P samples out of N total samples (where $P \ll N$) by adding perturbation Δ_i to the i^{th} training image. The perturbation is constrained in the ℓ_0 norm for the patch-based/backdoor trigger attack, or the ℓ_∞-norm in the case of optimization-based attacks we consider.

Attackers wish to find Δ so that a set of T target samples $(x_i^t, y_i^t)_{i=1}^T$ are classified with the

new, incorrect, adversarial labels y_i^{adv} after training by minimizing loss function $\mathcal{L}$:

$$\min_{\Delta \in \mathcal{C}} \sum_{i=1}^{T} \mathcal{L}\left(F(x_i^t, \theta(\Delta)), y_i^{\text{adv}}\right) \tag{4.1}$$

$$\theta(\Delta) \in_\theta \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(F(x_i + \Delta_i, \theta), y_i). \tag{4.2}$$

In this framework, we can understand *backdoor* attacks as choosing the optimal Δ directly based on a given rule (here via patch insertion onto training images with the target label y_i^{adv}), whereas optimization-based methods such as [Geiping et al., 2021] optimize some approximation of the (intractable) full bilevel optimization problem. Witches' Brew [Geiping et al., 2021] approximately optimizes Δ by modifying training data so the gradient of the training objective is aligned with the gradient of the adversarial loss $\mathcal{L}\left(F(x_i^t, \theta(\Delta)), y_i^{\text{adv}}\right)$, using optimization methods based on adversarial literature [Madry et al., 2017b, Chiang et al., 2020].

Both backdoor attacks and targeted data poisoning attacks rely upon the expressiveness of modern deep networks trained from scratch in order to "gerrymander" the network's decision boundary around specific target images [Geiping et al., 2021] — they behave normally on validation data and the chosen target is often made into a class outlier. As a result, changes in the training procedure, such as strong data augmentation, may have a significant impact on the success of poisoning by imposing regularity on the decision boundary and preventing target images from being gerrymandered into the wrong class [Schwarzschild et al., 2021]. Note the sensitivity of poisoning to changes in training has been confirmed by defenses involving gradient clipping/noising, which is the only defense shown in [Geiping et al., 2021, Hong et al., 2020] to degrade poisoning success. However, this defense ultimately proves impractical because of

	Poison Success (100%)	Val Accuracy (100%)	Poison Success (10%)	Val Accuracy (10%)
Baseline	100%	85%	57%	94%
mixup	100%	85%	42%	95%
CutMix	36%	94%	23%	95%

Table 4.1: Poison success and (clean) validation accuracy in the from-scratch setting against backdoor attacks. The first two columns correspond to poisoning all images in the target class, the last two columns correspond to poisoning 10% of images in the target class. All values are averaged over 4 runs.

the decreased natural accuracy that comes with robustness to poisoning. Thus, we aim to bridge this gap and develop small changes in training via data augmentation that defend against data poisoning without impeding normal training.

4.3 Method

Mixup can be interpreted as a method for convexifying class regions in the input space [Zhang et al., 2017a]. By enforcing that convex combinations of training points are assigned convex combinations of the labels, this augmentation method regularizes class boundaries, and removes small non-convex regions. In particular, we are motivated by the idea of using mixup to promote the removal of small "gerrymandered" regions in input space in which a target/poisoned data instance is assigned an adversarial label while being surrounded by (non-poisoned) instances with different labels.

In our experiments, we generalize the mixup process from [Zhang et al., 2017a] for mixture width k. Instead of a Beta distribution, convex coefficients are drawn from a Dirichlet distribution $\mathrm{Dir}[\alpha, \ldots, \alpha]$ of order k with interpolation parameter $\alpha = 1$.

We implement CutMix [Yun et al., 2019] as follows: Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ be the training dataset with $x_i \in \mathbb{R}^{w \times h \times c}$. Let (x_i, y_i) and (x_j, y_j) be two randomly sampled feature-target

Figure 4.1: An illustration of CutMix with poisoned data (truck) and non-poisoned data (deer). CutMix chops up the patch, lessening its visual impact in training data.

pairs. We randomly generate a box $M \in \{0,1\}^{w \times h}$ that indicates the pixels to be cut/pasted. All values of M are defined to be one except in a box centered at (r_x, r_y) where they are zero. To obtain the box location, we randomly sample the center $r_x \sim \text{Unif}(0, w)$ and $r_y \sim \text{Unif}(0, h)$. As in mixup, we use a coefficient $\lambda \sim \text{Dir}[1,1]$ to determine the relative contribution from each of the two randomly sampled data points. i.e. $r_w = w\sqrt{1-\lambda}$ and $r_h = h\sqrt{1-\lambda}$ give the width and height for the patch of zeros in M. The augmented image for CutMix becomes $\tilde{x} = M \bigodot x_i + (1 - M) \bigodot x_j$ with label $\tilde{y} = \lambda y_i + (1-\lambda) y_j$ obtained by mixing initial labels proportionally to the size of M. The binary operation $\bigodot$ represents element-wise multiplication. Cutout is similar, but the patch remains black. The cutout data point is given by $\tilde{x} = M \bigodot x_i$ with label y_i.

The procedure for MaxUp is taken from [Gong et al., 2020]: For each data point x_i in the original training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, a set $\{\tilde{x}_{i,j}\}_{j=1}^m$ of augmented data points are produced. Learning is defined according to a modified ERM,

$$\min_\theta \mathbb{E}_{x \sim \mathcal{D}}[\max_j \mathcal{L}(\tilde{x}_{i,j}, \theta)].$$

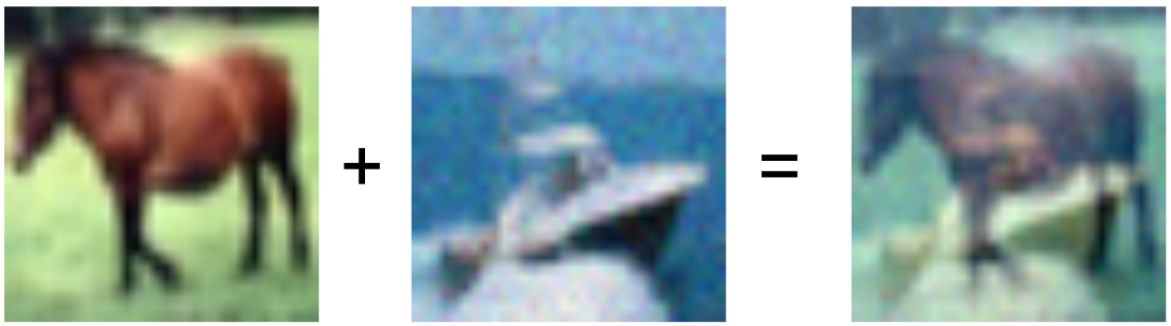

Figure 4.2: Mixup averages perturbed and clean images.

4.4 Experiments

4.4.1 Defending Against Backdoor Trigger Attacks

We first demonstrate the effectiveness of data augmentation at mitigating backdoor attacks while at the same time increasing test accuracy. To establish baselines for backdoor attacks, we train a ResNet-18 [He et al., 2016] on the CIFAR-10 dataset consisting of $50,000$ images in 10 balanced classes [Krizhevsky et al., 2009]. We insert triggers into the training set by adding 4×4 patches to training images in a randomly selected *target class*. We then evaluate on patched images from a new *base class* to see if the patched images are misclassified into the target class.

	Basic Attack (%)	Adaptive Attack (%)	Validation Accuracy (%)
Baseline	90.00	90.00	92.08
DP-SGD, 0.01	77.00	86.00	91.33
DP-SGD, 0.05	1.00	40.00	81.62
mixup	45.00	72.00	91.50
mixup (4-way)	5.00	55.00	87.45
CutMix	75.00	60.00	91.62
cutout	60.00	81.25	91.64
MaxUp-cutout	5.00	20.00	86.05

Table 4.2: Poison success and (clean) validation error results for the from-scratch setting against Witches' Brew. All values are averaged over 20 runs. Lower values in the first two columns are better. Higher numbers in the last column are better.

We perform this experiment in two different settings: when all training images in the base class are patched and when only 10% are patched. These two scenarios reflect varying access an adversary might have to their victim's training data.

We report the success of this attack at causing base images to be misclassified with the target label in Table 4.1. Although mixup data augmentation does not defend against the backdoor attack, CutMix dramatically reduces the success rate of the poison attack from 100% to 36% while simultaneously increasing validation accuracy by 9%. One explanation for the ineffectiveness of mixup in this domain is that, under this strategy, the base class can still be associated with the patch. On the other hand, CutMix randomly replaces parts of the images and therefore may cut patches apart. Moreover, CutMix improves clean validation accuracy since the network learns relevant features from the target class rather than simply relying on the patch.

4.4.2 Defending Against Targeted Poisoning Attacks

We now evaluate data augmentation as a defense against targeted poisoning attacks. To this end, we use the state-of-the-art method, Witches' Brew [Geiping et al., 2021], as well as an adaptive version in which attacks are generated on networks trained with the same data augmentations as the victim. We train a ResNet-18 on CIFAR-10, and we consider a threat model with ℓ_∞ bound $16/255$ and a budget of 1%. For all experiments, we consider a baseline model trained with random horizontal flips and random crops. We then compare modern data augmentation against differential-privacy based defenses with a special focus on practical clean validation accuracy. We average success over 20 runs, where each run consists of a randomly chosen target image and a randomly chosen adversarial label. In each run, a new model is trained from scratch

on the poisoned training set and evaluated on the target image. The lower the attack's success rate, the more effective the defense.

Table 4.2 shows that we can defend using differentially private SGD (DP-SGD) [Abadi et al., 2016] with sufficient amounts of Gaussian noise added to all training gradients — but this comes at a tremendous costs in validation accuracy. In this setting of an optimization-based attack, we also have to contend with another factor: *adaptive attacks*. An adaptive attack threat model assumes that the attacker is aware of the defense and can optimize their attack w.r.t to this defense. This attack is highlighted in the second column of Table 4.2. In this work, we adapt WitchesBrew to advanced data augmentation by incorporating the data augmentation into the training phase of the clean model used in the attack. This way poisoned data is created based on a model trained to be invariant to these augmentations. We adapt differential privacy as in [Geiping et al., 2021] by a straight-through estimate, finding it to be mitigated especially well by an adaptive attack.

In contrast to the loss of validation accuracy incurred by DP-SGD defenses, the defenses via data augmentation (lower rows in Table 4.2, blue dots in Figure 4.3) lose almost no validation accuracy. Nonetheless, they reduce poison success by up to 60% in the case of MaxUp based on four cutouts. Also notable is that for this case of optimization-based poisoning attacks, mixup seems to be the optimal data augmentation among those tried which do not degrade validation accuracy. We conjecture that this is due to the implicit linearity enforced between data points by mixup optimization, which makes it harder for the poisoning scheme to successfully generate outliers. We can even further increase the defensive capabilities of mixup by considering a four-fold mixture of images instead of the two-fold mixtures discussed in [Zhang et al., 2017a]. This defense is even stronger than vanilla mixup in both cases, but the four-fold mixtures of images

leads to data modifications so stark that they negatively influence validation accuracy.

In comparison to the backdoor trigger attacks, CutMix is less effective against the non-adaptive attack. However, we also find that for the adaptive setting, CutMix improves upon mixup. This is likely because the entire image perturbation appears in the mixed image when mixup is used, enabling reliable poisoning. In contrast, an adaptive attack on CutMix cannot *a priori* know the location of the cut patch, hence the attack is impeded even if it is known that CutMix is used.

4.5 Conclusions

Data poisoning attacks are increasingly threatening to machine learning practitioners. By and large, defenses have not kept pace with rapidly improved attacks. We demonstrate that modern data augmentation schemes can mitigate from-scratch poisoning while maintaining natural accuracy for both backdoor triggers and optimization-based attacks. We think these results suggest the possibility of specially-designed augmentations for poison defense, and we think this may be a fruitful direction for future research.

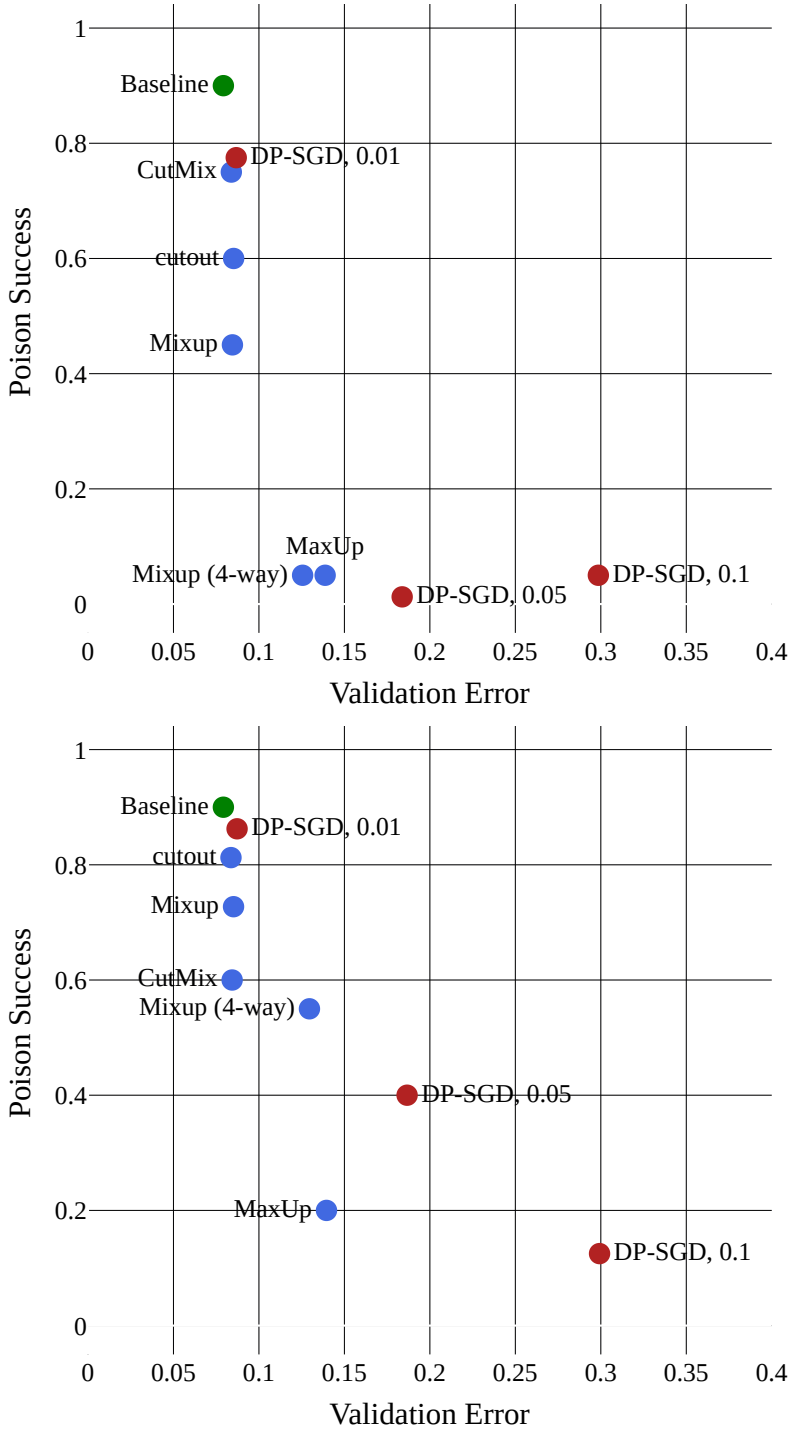

Figure 4.3: Poison success vs. (clean) validation accuracy for non-adaptive attacks (top) and adaptive attacks (bottom).

Chapter 5: A Deep Dive into Dataset Imbalance and Bias in Face Identification

Joint work with Steven Reich, Samuel Dooley, Hossein Souri, Micah Goldblum and Tom Goldstein. Accepted for publication at the AAAI/ACM Conference on AI, Ethics, and Society (AIES), 2023.

As the deployment of automated face recognition (FR) systems proliferates, bias in these systems is not just an academic question, but a matter of public concern. Media portrayals often center imbalance as the main source of bias, i.e., that FR models perform worse on images of non-white people or women because these demographic groups are underrepresented in training data. Recent academic research paints a more nuanced picture of this relationship. However, previous studies of data imbalance in FR have focused exclusively on the face *verification* setting, while the face *identification* setting has been largely ignored, despite being deployed in sensitive applications such as law enforcement. This is an unfortunate omission, as 'imbalance' is a more complex matter in identification; imbalance may arise in not only the training data, but also the testing data, and furthermore may affect the proportion of identities belonging to each demographic group *or* the number of images belonging to each identity. In this work, we address this gap in the research by thoroughly exploring the effects of each kind of imbalance possible in face identification, and discuss other factors which may impact bias in this setting.

5.1 Introduction

Automated face recognition is becoming increasingly prevalent in modern life, with applications ranging from improving user experience (such as automatic face-tagging of photos) to security (e.g., phone unlocking or crime suspect identification). While these advances are impressive achievements, decades of research have demonstrated disparate performance in FR systems depending on a subject's race [Phillips et al., 2011, Cavazos et al., 2020], gender presentation [Alvi et al., 2018, Albiero et al., 2020], age [Klare et al., 2012], and other factors. This is especially concerning for FR systems deployed in sensitive applications like law enforcement; incorrectly tagging a personal photo may be a mild inconvenience, but incorrectly identifying the subject of a surveillance image could have life-changing consequences. Accordingly, media and public scrutiny of bias in these systems has increased, in some cases resulting in policy changes.

One major source of model bias is dataset imbalance; disparities in rates of representation of different groups in the dataset. Modern FR systems employ neural networks trained on large datasets, so naturally much contemporary work focuses on what aspects of the training data may contribute to unequal performance across demographic groups. Some potential sources that have been studied include imbalance of the proportion of data belonging to each group [Wang and Deng, 2020, Gwilliam et al., 2021], low-quality or poorly annotated images [Dooley et al., 2021], and confounding variables entangled with group membership [Klare et al., 2012, Kortylewski et al., 2018, Albiero et al., 2020].

Dataset imbalance is a much more complex and nuanced issue than it may seem at first blush. While a naive conception of 'dataset imbalance' is simply as a disparity in the *number* of images per group, this disparity can manifest itself as either a gap in the number of identities per

group, or in the number of images per identity. Furthermore, dataset imbalance can be present in different ways in both the training and testing data, and these two source of imbalance can have radically different (and often opposite) effects on downstream model bias.

Past work has only considered the *verification* setting of FR, where testing consists of determining whether a pair of images belongs to the same identity. As such, 'imbalance' between demographic groups is not a meaningful concept in the test data. Furthermore, the distinction between imbalance of identities belonging to a certain demographic group versus that of images per identity in each demographic group has not been carefully studied in either the testing or the training data. All of these facets of imbalance are present in the face *identification* setting, where testing involves matching a probe image to a gallery of many identities, each of which contains multiple images. We illustrate this in Figure 5.1.

In this work, we unravel the complex effects that dataset imbalance can have on model bias for face identification systems. We separately consider imbalance (both in terms of identities or images per identity) in the train set and in the test set. We also consider the realistic social use case in which a large dataset is collected from an imbalanced population and then split at random, resulting in similar dataset imbalance in both the train and test set. We specifically focus on imbalance with respect to gender presentation, as (when restricting to only male- and female-identified individuals) this allows the proportion of data in each group to be tuned as a single parameter, as well as the availability of an ethically obtained identification dataset with gender presentation metadata of sufficient size to allow for subsampling without significantly degrading overall performance.

Our findings show that each type of imbalance has a distinct effect on a model's performance on each gender presentation. Furthermore, in the realistic scenario where the train and test

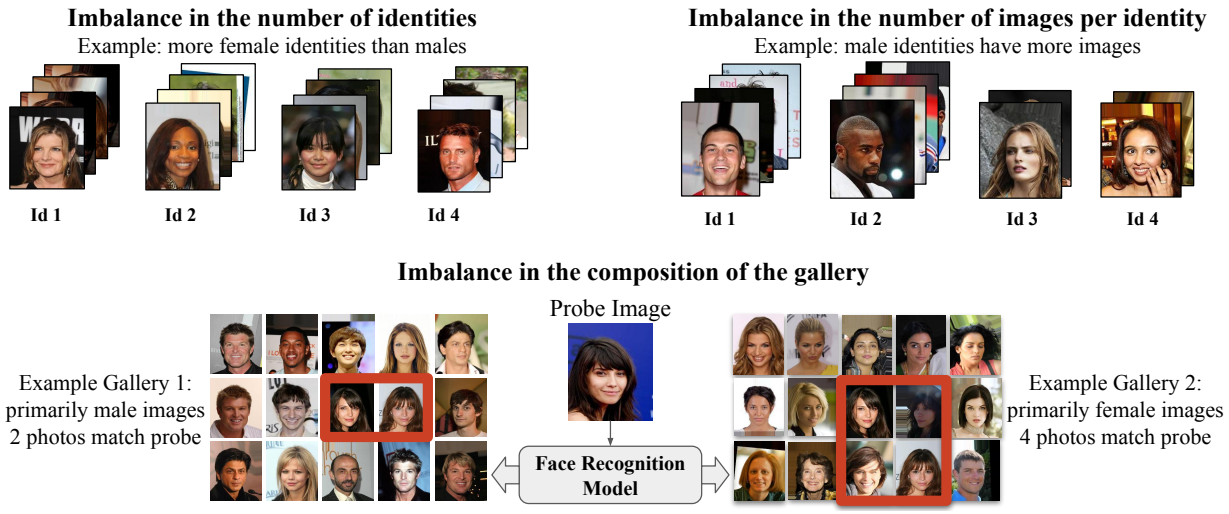

Figure 5.1: **Examples of imbalance in face identification**. Top left: data containing more female identities than male identities. Top right: data containing the same number of male and female identities, but more images per male identity. Bottom: two possible test (gallery) sets showing how the effects of different kinds of imbalance may interact.

set are similarly imbalanced, the train and test imbalance have the potential to interact in a way that leads to systematic underestimation of the true bias of a model during an audit. Thus any audit of model bias in face identification must carefully control for these effects.

The remainder of this chapter is structured as follows: Section 5.2 discusses related work, and Section 5.3 introduces the problem and experimental setup. Sections 5.4 and 5.5 give experimental results related to imbalance in the training set and test set, respectively, and Section 5.6 gives results for experiments where the imbalance in the training set and test set are identical. In Section 5.7.1, we evaluate randomly initialized feature extractors on test sets with various levels of imbalance to further isolate the effects of this imbalance from the effects of training. In Section 5.7.2, we investigate the correlation between the performance of models trained with various levels of imbalance and human performance.

5.2 Related Work

5.2.1 Imbalance in verification

Even before the advent of neural network-based face recognition systems, researchers have studied how the composition of training data affects verification performance. Phillips et al. [2011] compared algorithms from the Face Recognition Vendor Test [Phillips et al., 2009] and found that those developed in East Asia performed better on East Asian Faces, and those developed in Western countries performed better on Caucasian faces. Klare et al. [2012] expanded on these results by comparing performance across race, gender presentation, and age cohorts, observing that training exclusively on images of one demographic group improved performance on that group and decreased performance on the others. They further conclude that training on data that is “well distributed across all demographics" helps prevent extreme bias.

Multiple verification datasets have been proposed in the interest of eliminating imbalance as a source of bias in face verification. The *BUPT-BalancedFace* dataset [Wang and Deng, 2020] contains an approximately equal number of identities and images of four racial groups[1]. *Balanced Faces in the Wild* [Robinson et al., 2020] goes a step further, balancing identities and images across eight categories of race-gender presentation combinations. Also of note is the *BUPT-CBFace* dataset [Zhang and Deng, 2020], which is class-balanced (each identity possesses the same number of images), rather than demographically balanced.

Some recent work in verification has questioned whether perfectly balanced training data is in fact an optimal setting for reducing bias. Albiero et al. [2020] studied sources of bias along

[1]This work also introduces *BUPT-GlobalFace*, which instead approximately matches the distribution across races to that of the world population.

gender presentation; among their findings, they observe that balancing the amount of male and female training images and identities in the training data reduces, but does not eliminate, the performance gap between gender presentations. Similarly, Gwilliam et al. [2021] trained models on data with different racial makeups, finding that models which were trained with more images of African subjects had lower variance in performance on each race than those which were trained on balanced data.

5.2.2 Bias in Identification

Although the effect of imbalance on bias has only been explicitly studied in face verification, there is some research on identification which is relevant. The National Institutes of Standards and Technology performed large-scale testing of commercial identification algorithms, finding that many (though not all) exhibit gender presentation or racial bias [Grother et al., 2019]. The evaluators speculate that the training data or procedures contribute to this bias, but could not study this hypothesis due to the proprietary nature of the models. Dooley et al. [2021] evaluated commercial and academic models on a variant of identification in which each probe image is compared to 9 gallery images of distinct identities, but belonging to the same skin type and gender presentation. They find that academic models (and some, but not all, commercial models) exhibit skin type and gender presentation bias despite a testing regime which makes imbalance effectively irrelevant.

5.2.3 Imbalance in Deep Learning

Outside the realm of facial recognition, there is much study about the impacts of class imbalance in deep learning. In standard machine learning techniques, i.e., non-deep learning, there are many well-studied and proven techniques for handling class imbalances like data-level techniques [Van Hulse et al., 2007, Chawla et al., 2002, 2004], algorithm-level methods [Elkan, 2001, Ling and Sheng, 2008, Krawczyk, 2016], and hybrid approaches [Chawla et al., 2003, Sun et al., 2007, Liu et al., 2008]. In deep learning, some take the approach of random over or under sampling [Hensman and Masko, 2015, Lee et al., 2016, Pouyanfar et al., 2018]. Other methods adjust the learning procedure by changing the loss function [Wang et al., 2016] or learning cost-sensitive functions for imbalanced data [Khan et al., 2017]. We refer the reader to Buda et al. [2018], Johnson and Khoshgoftaar [2019], for a thorough review of deep learning-based imbalance literature. Much of the class-imbalance work has been on computer vision tasks, though generally has not examined specific analyses like we present in this work like network initialization, face identification, or intersectional demographic imbalances.

5.2.4 Other sources of bias in facial recognition

Face recognition is a complex, sociotechnical system where biases can originate from the algorithms [Danks and London, 2017], preprocessing steps [Dooley et al., 2022], and human interpretations [Chouldechova and Roth, 2020]. While we do not explicitly examine these sources, we refer the reader to Mehrabi et al. [2021], Suresh and Guttag [2019] for a broader overview of sources of bias in machine learning.

5.3 Face Identification Setup

Face recognition has two tasks: face verification and face identification. The first refers to verifying whether a person of interest (called the *probe image*) and a person in a reference photo are the same. This is the setting that might be applied, e.g., to phone unlocking or other identity confirmation. In contrast, face identification involves matching a probe image against a set of images (called the *gallery*) with known identities. This application is relevant to search tasks, such as identifying the subject of a photo from a database of driver's license or mugshot photos.

In a standard face recognition pipeline, an image is generally first pre-processed by a face detection system which may serve to locate and align target faces to provide more standardized images to the recognition model. State-of-the-art face recognition models exploit deep neural networks which are trained on large-scale face datasets for a classification task. At test time, the models work as feature extractors, so that the similarity between a probe image and reference photo (in verification) or gallery photos (in identification) is computed in the feature space. In verification, the similarity score is then compared with a predefined threshold, while in identification a k-nearest neighbors search is performed using the similarity scores with the gallery images.

We focus on the face identification task in our experiments and explore how different kinds of data balance affect the models performance across demographic groups (specifically, the disparity in performance on male and female targets). We also analyze how algorithmic bias correlates with human bias on InterRace, a manually curated dataset specifically designed for bias auditing, with challenging face recognition questions and provided annotations for gender presentation and skin color [Dooley et al., 2021].

Our experiments use state-of-the-art face recognition models. We train MobileFaceNet [Chen et al., 2018], ResNet-50, and ResNet-152 [He et al., 2016] feature extractors each with a CosFace and ArcFace head which improve the class separability of the features by adding angular margin during training [Deng et al., 2019, Wang et al., 2018]. For training and evaluation we use the CelebA dataset [Liu et al., 2015], which provides annotations for gender presentation. As our main research questions focus on the impact of class imbalance, we pay special attention to the balance of the gender presentation attribute in our training. The original dataset contains more female identities. As such, we create a balanced training set containing 140,000 images from 7,934 identities with equal number of identities and total number of images from each gender presentation. We also create a perfectly balanced test set containing 14,000 images from 812 identities. The identities in the train and test sets are disjoint. We call these the *default train* and *default test* sets. All models are trained with class-balanced sampling to ensure equal contribution of identities to the loss. We additionally include results for models trained without over-sampling in Section 5.9.1.

Recall that our research question is to investigate how class imbalances affect face identification. In order to answer this question, we train models on a range of deliberately imbalanced subsamples of the default training set, and test models on a range of deliberately imbalanced subsamples of the default test set, in order to explore the impact on the model's performance for each gender presentation.

To evaluate the models, we compute rank-1 accuracy over the test set. Specifically, for each test image we treat the rest of the test set as gallery images and find if the closest gallery image in the feature space (as defined by cosine similarity) of a model is an image of the same person.

When we make comparisons with human performance (Section 5.7.2), we use the InterRace

Table 5.1: Details on the number of identities, total number of images and average number of images per identity used in experiments with train and test data balance. We also report statistics for the default train and test sets. M denotes male, F denotes female.

Setting	M ids	F ids	Total M img	Total F img	M img/id	F img/id	Total id	Total img
Train default	3967	3967	70k	70k	17.65	17.65	7934	140k
Train id balance	0 - 3967	0 - 3967	0 - 70k	0 - 70k	17.65	17.65	3967	70k
Train img balance	3967	3967	14k - 56k	14k - 56k	3.53 - 14.11	3.53 - 14.11	7934	70k
Test default	406	406	7k	7k	17.24	17.24	812	14k
Test id balance	0 - 406	0 - 406	0 - 7k	0 - 7k	17.24	17.24	406	7k
Test img balance	406	406	1.4k - 5.6k	1.4k - 5.6k	3.45 - 13.80	3.45 - 13.80	812	7k

dataset [Dooley et al., 2021]. Since the InterRace dataset is derived from both the CelebA and LFW [Huang et al., 2007] datasets, we additionally train models on the InterRace-train split of CelebA, containing images of identities not included in the InterRace dataset. Similar to other experiments, we train models with varying levels of either identity and image imbalance.

5.4 Balance in the Train Set

5.4.1 Balancing the number of identities

Experiment Description. To explore the effect of train set balance in the number of identities on gender presentation bias, we construct train data splits with different ratios of female and male identities, while ensuring that the average number of images per identity is the same across gender presentations. Therefore, in all splits we have the same total number of images and total number of identities, but the proportion of female and male identities varies. We consider splits with $0:10$, $1:9$, $2:8$, ..., $10:0$ ratios, each having 70,000 total images from 3967 identities. We evaluate the models on the (perfectly balanced) default test set and report rank-1 face identification accuracy as described in Section 5.3. More details of train set splits can be found in Table 5.1.

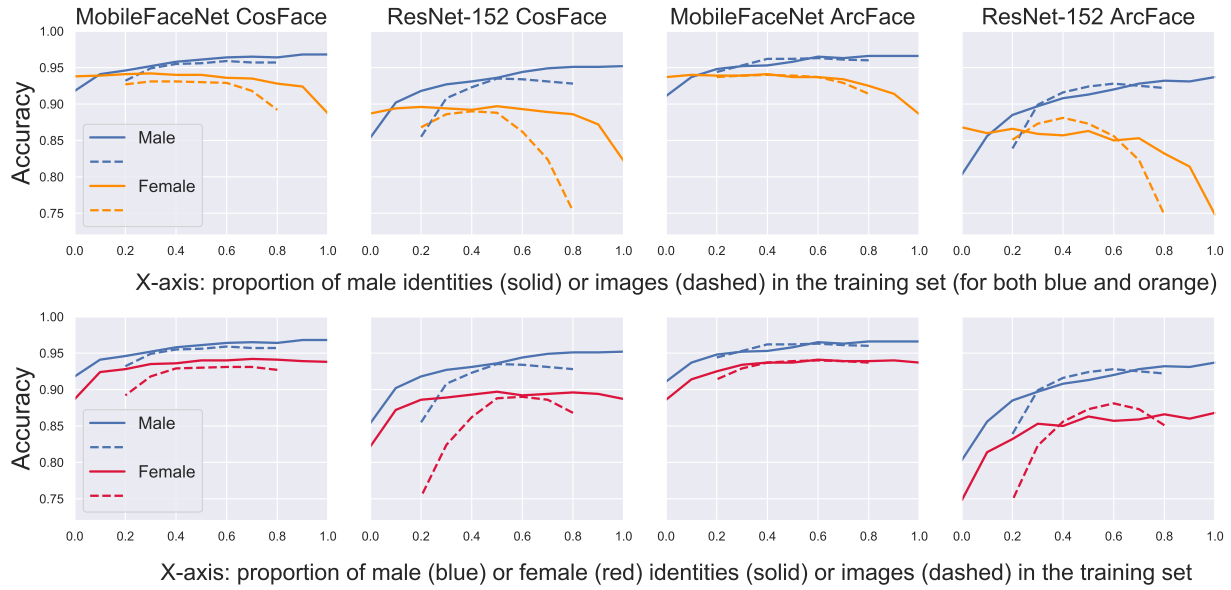

Figure 5.2: **Train Set Imbalance.** Results of experiments that change the train set gender presentation balance. Top row: male and female accuracy are plotted against the proportion of male data in the train set. Bottom row: for an alternate view, female accuracy is flipped horizontally, so that it is plotted against the proportion of *female* data in the train set. All models are tested on the default balanced test set.

Results. We compute accuracy scores separately for male and female test images for models trained on each of the train splits and depict them in Figure 5.2 with solid lines. From the first row plots, we observe that a higher proportion of male identities in the train set leads to an increase in male accuracy and decrease in female accuracy, with the most significant drops occurring near the extreme $10:0$ imbalance. This indicates that it is very important to have at least a few identities from the target demographic group in the train set; once the representation of the minority group reaches 10%, the marginal gain of additional identities becomes less. We also observe that for most models, the female accuracy drops slightly when the proportion of female identities exceeds 80% of the training data, which does not happen to the male group. Consult Table 5.2 for the numerical results.

Regarding the model architectures, MobileFaceNet models trained with both CosFace and ArcFace heads outperform ResNet models on both female and male images and have smaller absolute accuracy gap. However, the error ratio is similar across the models, see Table 5.2. Finally, the accuracy gap is closed for all models when the train set consists of about 10% male and 90% female identities.

In addition, in the second row of Figure 5.2 we compare how similar these trends are for females and males by plotting female accuracy against the proportion of *female* identities in the train set. One can see that for MobileFaceNet models the accuracy on male and female images increases similarly when increasing the proportion of "target" identities up to 80%. However, for ResNet models adding more female identities in the train set results in smaller gains compared to the effect of adding more male identities on male accuracy.

5.4.2 Balancing the number of images per identity

In the previous subsection, we fixed the average number of images per identity in each gender presentation and adjusted the number of identities. We now will do the reverse: fix the number of identities and vary the images per identity.

Experiment Description. We change the average number of images per male and female identity, but fix the number of identities of each gender presentation. We consider ratios $2:8$, ..., $8:2$, each having $70,000$ images from $7,934$ identities. We do not consider more extreme ratios, which would result in identities with fewer than 3 images.

Results.

The dashed lines in Figure 5.2 illustrate the accuracy of the models trained on described

data splits. From the first row plots we see that, similar to the previous experiment, increasing the number of male images in the train set leads to increased accuracy on male and decreased accuracy on female images. Interestingly, we observe a decrease in performance for both demographic groups when the images of that group constitute more than 60% of train data; this is most easily visible in the second row of Figure 5.2. However, we find that this effect results from the widely used class-balanced sampling training strategy, and models trained without the default oversampling are more robust to imbalance in the number of images per identity, see details in Section 5.9.1 and Figure 5.8. The “fair point" where female accuracy is closest to male accuracy occurs when around 20% of images are of males.

When comparing the effect of imbalance in the number of identities and the number of images per identity (solid and dashed lines respectively in Figure 5.2), we see that ResNet models are more susceptible to image imbalance than to identity imbalance, which is also a phenomenon specific to the common class-balanced sampling.

5.5 Balance in the Test Set

5.5.1 Balancing the number of identities

Experiment Description. Analogous to the train set experiments, we split the test data (the gallery) with different ratios of female and male identities, while keeping the same average number of images per identity for both demographic groups. For each ratio, we split the test data with 5 random seeds and report average rank-1 accuracy of the models trained on default train data. The results are shown in the solid lines of Figure 5.3, as well as in Table 5.4.

Results. We observe that increasing the proportion of identities of a target demographic

Figure 5.3: **Test Set Imbalance.** Results of experiments that change the test set gender presentation balance. Top row: male and female accuracy are plotted against the proportion of male data in the test set. Bottom row: for an alternate view, female accuracy is flipped horizontally, so that it is plotted against the proportion of *female* data in the test set. All models are trained on the default balanced train set. For each experiment, the test set was split with 5 random seeds, and the results are averaged across seeds.

group in the test set hurts the model's performance on that demographic group, and this trend is similar for male and female images. Intuitively, this is because face recognition models rarely match images to one of a different demographic group; therefore by adding more identities of a particular demographic group, we add more potential false matches for images from that demographic group, which leads to higher error rates. We also see that ResNet models are more sensitive to the number of identities in the gallery set than MobileFaceNet models.

5.5.2 Balancing the number of images per identity

Experiment Description. Now, we investigate how increasing or decreasing the number of images per identity affects the performance and bias of the models. Again, we split the test

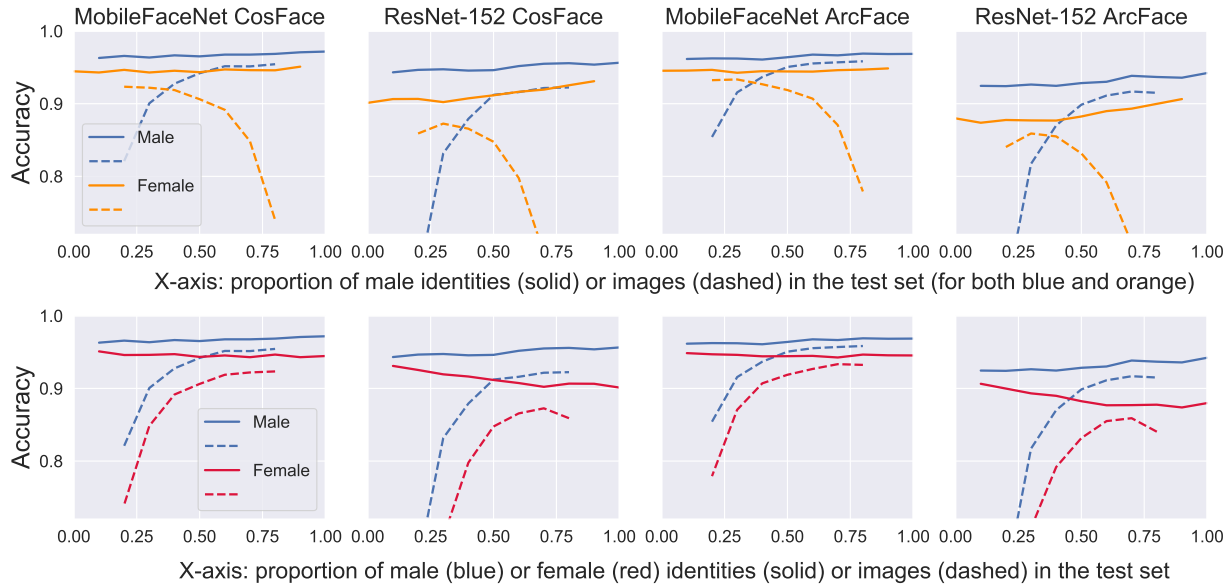

Figure 5.4: **Train & Test Set Imbalance.** Results of experiments that adjust the gender presentation balance in both the train and test set. Top row: male and female accuracy are plotted against the proportion of male data used in both the train and test set. Bottom row: for an alternate view, female accuracy is flipped horizontally, so that it is plotted against the proportion of *female* data in both the train and test set. For each experiment, the test set was split with 5 random seeds, and the results are averaged across seeds.

sets with different ratios of total number of images across gender presentations, but same number of identities, each with 5 random seeds. These results are recorded as dashed lines in Figure 5.3, as well as in Table 5.5.

Results. Unlike the results with identity balance, increasing the average number of images per identity leads to performance gains, since this increases the probability of a match with an image of the same person. Also, image balance affects the performance more significantly than identity balance, and these trends are similar across all the models and both gender presentations. Finally, we note that the “fair point" for image balance in the test set occurs at about 30% male images; contrast this with identity balance, for which no fair point appears to exist.

5.6 A cautionary tale: matching the balance in the train and gallery data

Using our findings from above, we conclude that common machine learning techniques to create train and test splits can lead to Simpson's paradoxes which lead to a false belief that a model is unbiased. It is standard practice to make random train/test splits of a dataset. If the original dataset is imbalanced, as is commonly the case, the resulting splits will be imbalanced in similar ways. As we have seen above, the effects of imbalance in the train and test splits may oppose one another, causing severe underestimation of model bias when measured using the test split. This occurs because the minority status of a group in the train split will bias the model towards low accuracy on that group, while the correspondingly small representation in the test split will cause an increase in model accuracy, partially or entirely masking the true model bias. The results for these experiments are presented in Figure 5.4 and Tables 5.6, 5.7.

Balancing the number of identities We create train and test sets with identical distributions of identities. Recalling the results from prior experiments, increasing the number of identities for the target group in the training stage improves accuracy on that group, while adding more identities in the gallery degrades it. Interestingly, when we increase the proportion of male identities in *both* train and test sets, we observe gains in both male and female accuracy, and that trend is especially strong for ResNet models.

Balancing the number of images per identity Having more images is beneficial in both train and test stages. Therefore, the effect of image balance is amplified when both train and test sets are imbalanced in a similar way. Similar to the train set experiments, having more than 70% female images in both train and test sets leads to slight drops in female accuracy on ResNet models, which again is a result of the default class-balanced oversampling strategy.

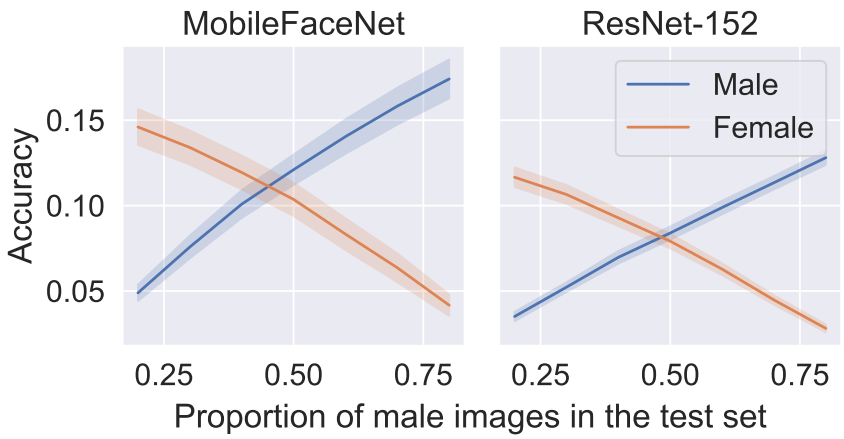

Figure 5.5: **Random Feature Extractors.** The plot illustrates male (blue) and female (orange) accuracy of random feature extractors against the proportion of male images in the test set. The standard deviation is computed across 10 random initializations.

5.7 Bias comparisons

We ask two concluding questions: one about whether class imbalance captures all the inherent bias and the other about how the bias we see compares to human biases. First, we explore how data imbalances cause biases in random networks and find surprising conclusions. Then, we ask how class imbalances in machines compare to how humans exhibit bias on face identification tasks.

5.7.1 Bias in random feature extractors

Given a network with random initializations, we would expect that evaluation on a balanced test set would result in equal performance on males and females, and likewise that male performance on a set with a particular proportion of male identities would be the same as female performance when that proportion is reversed. However, this is not the case. We test randomly

initialized feature extractors on galleries with varying levels of image imbalance. Figure 5.5 summarizes the results of these experiments. We observe that both models have higher male performance when the test set is perfectly balanced, and that performance on males is higher when they make up 80% of the test set than female performance when they make up 80% of the test set. This provides strong evidence that there are sources of bias that lie outside what we explore here and which are potential confounders to a thorough study of bias in face identification; further work on this is warranted.

5.7.2 Are models biased like humans?

Numerous psychological and sociological studies have identified gender, racial, and other biases in human performance on face recognition tasks. Dooley et al. [2021] studied whether humans and FR models exhibit similar biases. They evaluated human and machine performance on the curated InterRace test questions, and found models indeed tend to perform better on the same groups as, and with comparable gender presentation bias ratios to, humans. In this section, we use their human survey data to explore two related questions: how correlated are model and human performance *at the question level*, and how does this change with different levels of imbalance in training data?

To answer these questions, we define a metric which allows us to distinguish how well a model performs on each InterRace identification question. Let

$$L_2 \text{ ratio} = \frac{\|v_{probe} - v_{false}\|_2}{\|v_{probe} - v_{true}\|_2 + \|v_{probe} - v_{false}\|_2},$$

where v_{probe}, v_{true}, v_{false} are the feature representations of the probe image, the correct gallery

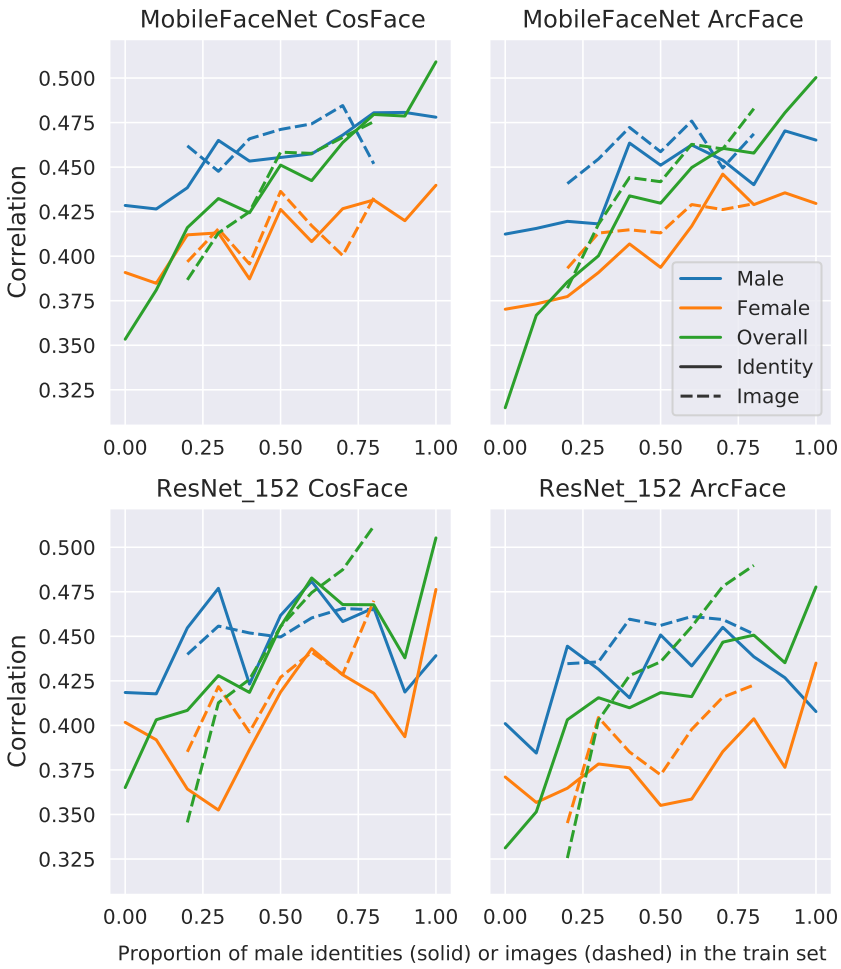

Figure 5.6: Pearson correlation of $L2$ ratio vs. human accuracy for various models as proportion of male training data varies.

image, and the nearest incorrect gallery image, respectively.[2] This value is 1 when the probe and correct image's representations coincide, 0 when the probe and incorrect image's representations coincide and 0.5 when the probe's representation is equidistant from those of the correct and incorrect image. Figure 5.7 depicts examples of scatterplots comparing model confidence to human accuracy on each InterRace question.

Figure 5.6 shows the correlation between $L2$ ratio and human performance for various

[2]We note that other measures of confidence in a k-nearest neighbors setting, such as those discussed in [Dalitz, 2009], are inappropriate for this application.

models at each of the training imbalance settings that we have considered in earlier experiments. We see that the correlation between these values over *all* questions tends to rise as the proportion of male training data increases. However, the correlation when separately considering male and female questions does not rise as monotonically, or as much, from left to right as the overall correlation does. This suggests that the correlation between human and machine performance is largely driven by the fact that models and humans both find identifying females more difficult than identifying males, and that this disparity is exacerbated when the model in question is trained on male-dominated data. On the other hand, the *particular* males and females that are easier or harder to identify appear to differ between models and humans, which suggests the *reasons* for bias in humans and machines are different.

5.8 Actionable Insights

We note five actionable insights for machine learning engineers and other researchers from this work. First, **overrepresenting the target demographic group can sometimes hurt that group**. Sometimes having more balanced data is the key. Also, class-balanced sampling might hurt representation learning when the data is not balanced with respect to the number of images per identity. Second, **gallery set balance is as important as train set balance**, contrary to how face verification class imbalances work. Third, **having the same distribution of identities and average number of images per identity is not an unbiased way to evaluate a model**, since the effects of balance in train and test sets can be amplified (in case of images) or cancel each other (in case of identities). Fourth, **train and test class imbalances are not the only cause of bias** in face identification evaluation since even random models do not perform equally poorly

on female and male images. Finally, even though both humans and machine find female images more difficult to recognize, **it seems that the reasons for bias are different in people and models**. We know that this work sheds light on common mistakes in bias computations for many facial recognition tasks and hope that auditors and engineers will incorporate our insights into their methods.

5.9 Experimental Details and Additional Results

5.9.1 Results for models trained without class-balanced sampling.

To explore the effect of class-balanced sampling on the results of our experiments, we train additional models without any oversampling strategies. Figures 5.8 - 5.10 show results of our experiments for MobileFaceNet and ResNet-152 models trained without oversampling. We find that most trends are similar to ones observed in the models trained with class-balanced sampling, however models trained without oversampling are more robust to balance in the number of images per identity, see Figure 5.8. In particular, the effect of balancing the number of images (dashed lines) is similar to the effect of balancing the number of identities (solid lines) for all models, but ResNet-152 trained with ArcFace head. This leads us to a conclusion that using class-balanced sampling strategy is not beneficial in scenarios of severe imbalance in number of images per identity in face recognition models.

5.9.2 Additional Plots and Tables

Figure 5.11 shows the results of the train set imbalance experiment when evaluated on the InterRace test set. Tables 5.2 - 5.7 precisely detail the number of male and female identities and images used in each experiment, as well as the accuracy on male and female targets and the female-to-male error ratio.

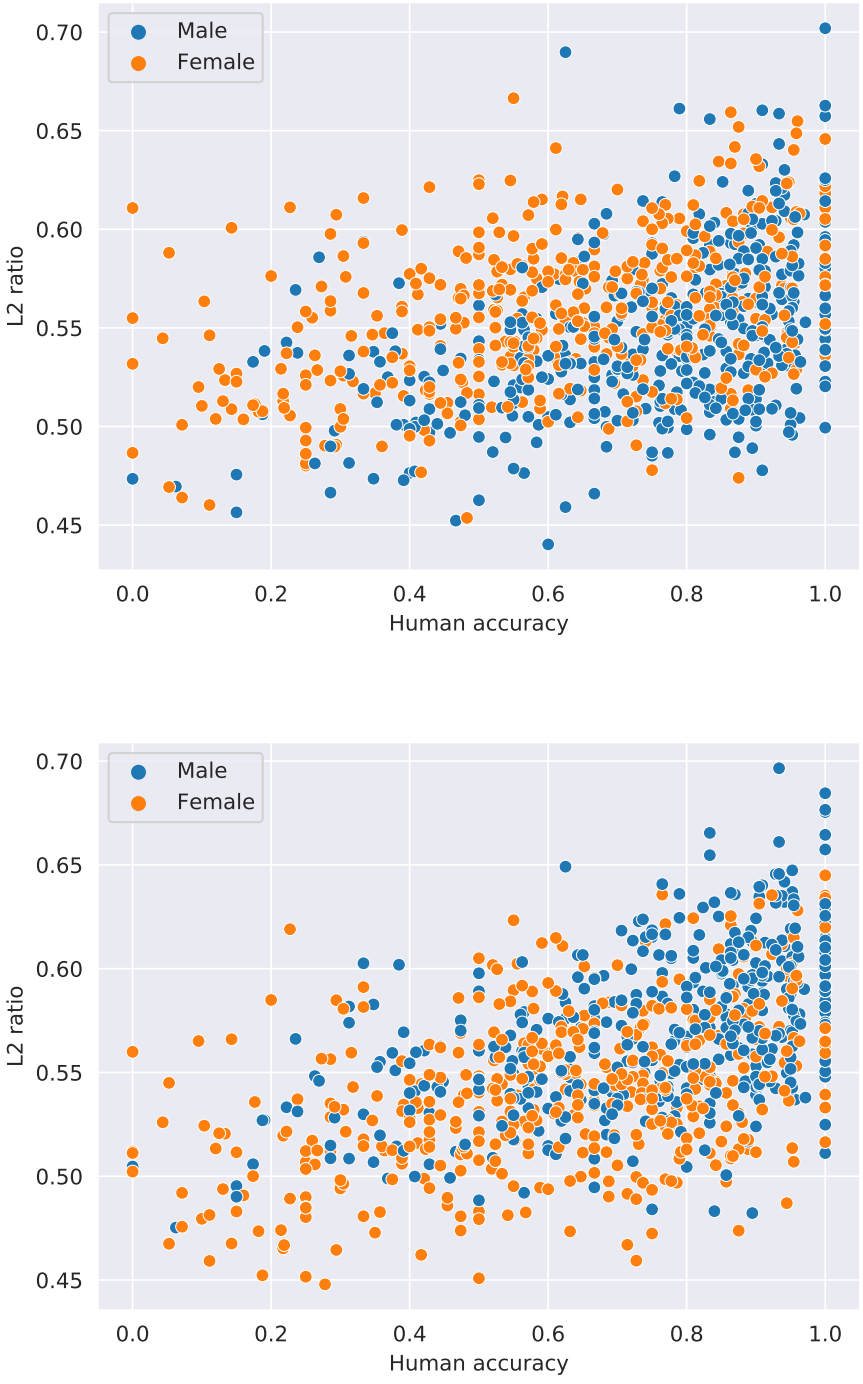

Figure 5.7: Scatterplots of model L2 ratio vs. human accuracy on each question in the InterRace identification dataset. Both models are MobileFaceNets trained with CosFace loss. (Left) a model trained on exclusively female images. (Right) a model trained on exclusively male images.

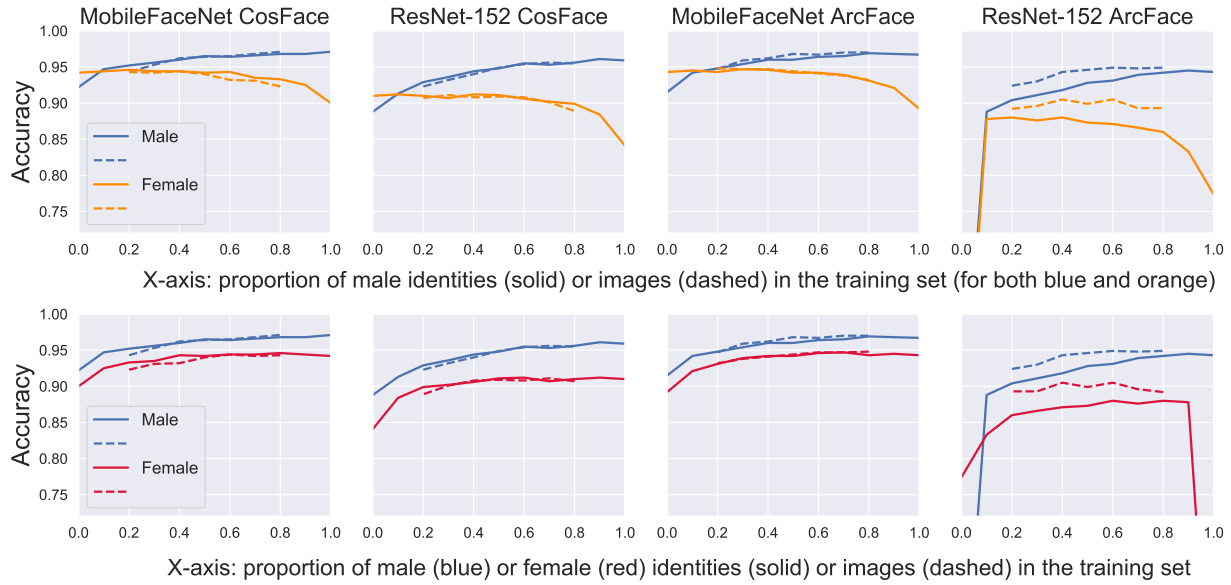

Figure 5.8: **Train Set Imbalance**. Results of experiments that change the train set gender presentation balance for MobileFaceNet and ResNet-152 models trained **without class-balanced sampling**.

Figure 5.9: **Test Set Imbalance.** Results of experiments that change the test set gender presentation balance for MobileFaceNet and ResNet-152 models trained **without class-balanced sampling**.

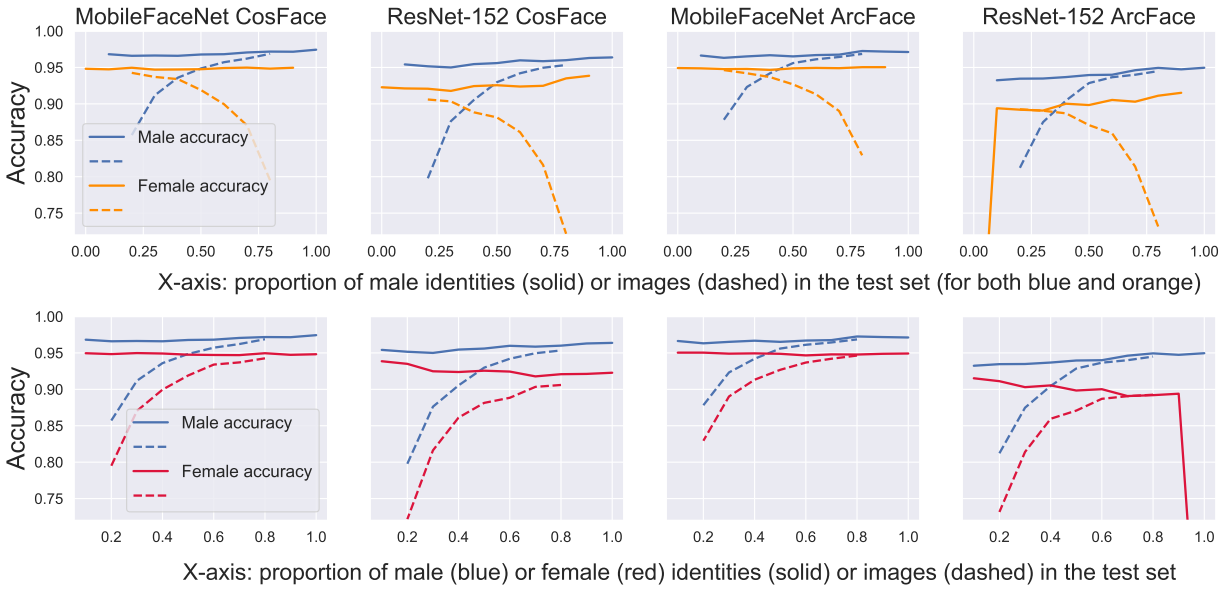

Figure 5.10: **Train & Test Set Imbalance.** Results of experiments that adjust the gender presentation balance in both the train and test set for MobileFaceNet and ResNet-152 models trained **without class-balanced sampling**.

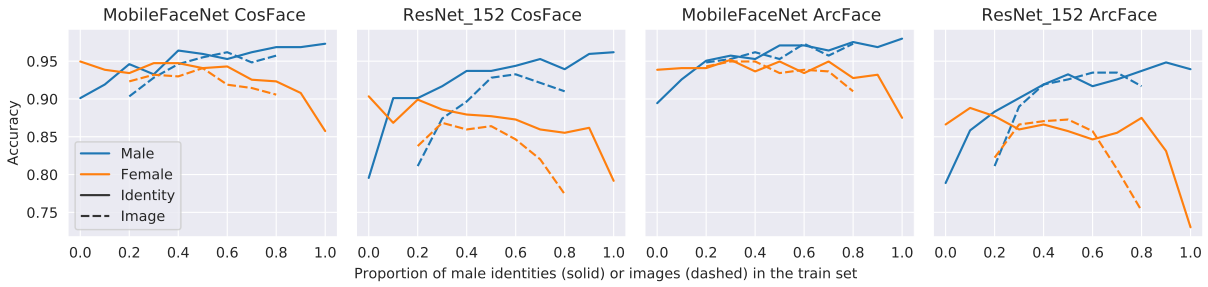

Figure 5.11: **Train Set Imbalance.** Results of experiments testing models trained with different gender presentation balance on the InterRace dataset. These plots are analogous to the first row of Figures 5.2.

Table 5.2: **Train Set Id Imbalance.** The female and male accuracy computed over the default balanced test set for models trained on data with various ratios of number of male and female identities. See details of the experiment in Section 5.4.1

Model	Ids Ratio	M ids	F ids	M imgs	F imgs	M acc	F acc	Error Ratio
MFN CosFace	0 : 10	0	3967	0	70k	0.918	0.938	0.76
	1 : 9	397	3570	7k	63k	0.941	0.939	1.03
	2 : 8	793	3174	14k	56k	0.946	0.941	1.09
	3 : 7	1190	2777	21k	49k	0.952	0.942	1.21
	4 : 6	1587	2380	28k	42k	0.958	0.940	1.43
	5 : 5	1984	1984	35k	35k	0.961	0.940	1.54
	6 : 4	2380	1587	42k	28k	0.964	0.936	1.78
	7 : 3	2777	1190	49k	21k	0.965	0.935	1.86
	8 : 2	3174	793	56k	14k	0.964	0.928	2.00
	9 : 1	3570	397	63k	7k	0.968	0.924	2.37
	10 : 0	3967	0	70k	0	0.968	0.887	3.53
MFN ArcFace	0 : 10	0	3967	0	70k	0.911	0.937	0.71
	1 : 9	397	3570	7k	63k	0.937	0.940	0.95
	2 : 8	793	3174	14k	56k	0.948	0.939	1.17
	3 : 7	1190	2777	21k	49k	0.952	0.939	1.27
	4 : 6	1587	2380	28k	42k	0.953	0.941	1.26
	5 : 5	1984	1984	35k	35k	0.958	0.937	1.50
	6 : 4	2380	1587	42k	28k	0.965	0.937	1.80
	7 : 3	2777	1190	49k	21k	0.963	0.934	1.78
	8 : 2	3174	793	56k	14k	0.966	0.925	2.21
	9 : 1	3570	397	63k	7k	0.966	0.914	2.53
	10 : 0	3967	0	70k	0	0.966	0.886	3.35
ResNet-152 CosFace	0 : 10	0	3967	0	70k	0.854	0.887	0.77
	1 : 9	397	3570	7k	63k	0.902	0.894	1.08
	2 : 8	793	3174	14k	56k	0.918	0.896	1.27
	3 : 7	1190	2777	21k	49k	0.927	0.894	1.45
	4 : 6	1587	2380	28k	42k	0.931	0.892	1.57
	5 : 5	1984	1984	35k	35k	0.936	0.897	1.61
	6 : 4	2380	1587	42k	28k	0.944	0.893	1.91
	7 : 3	2777	1190	49k	21k	0.949	0.889	2.18
	8 : 2	3174	793	56k	14k	0.951	0.886	2.33
	9 : 1	3570	397	63k	7k	0.951	0.872	2.61
	10 : 0	3967	0	70k	0	0.952	0.822	3.71
ResNet-152 ArcFace	0 : 10	0	3967	0	70k	0.803	0.868	0.67
	1 : 9	397	3570	7k	63k	0.856	0.860	0.97
	2 : 8	793	3174	14k	56k	0.885	0.866	1.17
	3 : 7	1190	2777	21k	49k	0.897	0.859	1.37
	4 : 6	1587	2380	28k	42k	0.908	0.857	1.55
	5 : 5	1984	1984	35k	35k	0.913	0.863	1.57
	6 : 4	2380	1587	42k	28k	0.920	0.850	1.88
	7 : 3	2777	1190	49k	21k	0.928	0.853	2.04
	8 : 2	3174	793	56k	14k	0.932	0.832	2.47
	9 : 1	3570	397	63k	7k	0.931	0.814	2.70
	10 : 0	3967	0	70k	0	0.937	0.748	4.00

Table 5.3: **Train Set Img Imbalance.** The female and male accuracy computed over the default balanced test set for models trained on data with various ratios of number of images per male and female identity. See details of the experiment in Section 5.4.2

Model	Img Ratio	# M ids	# F ids	# M imgs	# F imgs	M Acc	F Acc	Error Ratio
MFN CosFace	2 : 8	3967	3967	14k	56k	0.932	0.927	1.07
	3 : 7	3967	3967	21k	49k	0.949	0.931	1.35
	4 : 6	3967	3967	28k	42k	0.955	0.931	1.53
	5 : 5	3967	3967	35k	35k	0.956	0.930	1.59
	6 : 4	3967	3967	42k	28k	0.959	0.929	1.73
	7 : 3	3967	3967	49k	21k	0.957	0.918	1.91
	8 : 2	3967	3967	56k	14k	0.957	0.892	2.51
MFN ArcFace	2 : 8	3967	3967	14k	56k	0.944	0.937	1.13
	3 : 7	3967	3967	21k	49k	0.953	0.939	1.30
	4 : 6	3967	3967	28k	42k	0.962	0.940	1.58
	5 : 5	3967	3967	35k	35k	0.962	0.939	1.61
	6 : 4	3967	3967	42k	28k	0.963	0.937	1.70
	7 : 3	3967	3967	49k	21k	0.961	0.929	1.82
	8 : 2	3967	3967	56k	14k	0.960	0.914	2.15
ResNet-152 CosFace	2 : 8	3967	3967	14k	56k	0.855	0.868	0.91
	3 : 7	3967	3967	21k	49k	0.908	0.886	1.24
	4 : 6	3967	3967	28k	42k	0.923	0.890	1.43
	5 : 5	3967	3967	35k	35k	0.935	0.888	1.72
	6 : 4	3967	3967	42k	28k	0.934	0.862	2.09
	7 : 3	3967	3967	49k	21k	0.931	0.824	2.55
	8 : 2	3967	3967	56k	14k	0.928	0.753	3.43
ResNet-152 ArcFace	2 : 8	3967	3967	14k	56k	0.839	0.851	0.93
	3 : 7	3967	3967	21k	49k	0.899	0.873	1.26
	4 : 6	3967	3967	28k	42k	0.916	0.881	1.42
	5 : 5	3967	3967	35k	35k	0.924	0.873	1.67
	6 : 4	3967	3967	42k	28k	0.928	0.856	2.00
	7 : 3	3967	3967	49k	21k	0.925	0.823	2.36
	8 : 2	3967	3967	56k	14k	0.922	0.748	3.23

Table 5.4: **Test Set Id Imbalance.** The female and male accuracy for models trained on default train set computed on test set with various ratios of number of male and female identities. See details of experiment in Section 5.5.1.

Model	Ids Ratio	# M ids	# F ids	# M imgs	# F imgs	M Acc	F Acc	Error Ratio
MFN CosFace	0 : 10	0	406	0	7000	-	0.961	-
	1 : 9	41	365	700	6300	0.983	0.961	2.25
	2 : 8	81	325	1400	5600	0.981	0.960	2.04
	3 : 7	122	284	2100	4900	0.981	0.960	2.09
	4 : 6	162	244	2800	4200	0.981	0.962	2.00
	5 : 5	203	203	3500	3500	0.980	0.961	1.95
	6 : 4	244	162	4200	2800	0.980	0.963	1.83
	7 : 3	284	122	4900	2100	0.979	0.964	1.77
	8 : 2	325	81	5600	1400	0.979	0.969	1.45
	1 : 9	365	41	6300	700	0.978	0.962	1.72
	0 : 10	406	0	7000	0	0.978	-	-
MFN ArcFace	0 : 10	0	406	0	7000	-	0.959	-
	1 : 9	41	365	700	6300	0.980	0.959	2.07
	2 : 8	81	325	1400	5600	0.980	0.960	1.98
	3 : 7	122	284	2100	4900	0.981	0.958	2.17
	4 : 6	162	244	2800	4200	0.981	0.960	2.05
	5 : 5	203	203	3500	3500	0.979	0.961	1.89
	6 : 4	244	162	4200	2800	0.979	0.963	1.81
	7 : 3	284	122	4900	2100	0.979	0.962	1.84
	8 : 2	325	81	5600	1400	0.979	0.968	1.50
	1 : 9	365	41	6300	700	0.977	0.963	1.58
	0 : 10	406	0	7000	0	0.978	-	-
ResNet-152 CosFace	0 : 10	0	406	0	7000	-	0.944	-
	1 : 9	41	365	700	6300	0.981	0.943	2.94
	2 : 8	81	325	1400	5600	0.979	0.945	2.58
	3 : 7	122	284	2100	4900	0.977	0.946	2.37
	4 : 6	162	244	2800	4200	0.977	0.947	2.28
	5 : 5	203	203	3500	3500	0.974	0.947	2.01
	6 : 4	244	162	4200	2800	0.974	0.949	1.99
	7 : 3	284	122	4900	2100	0.974	0.952	1.87
	8 : 2	325	81	5600	1400	0.973	0.957	1.59
	1 : 9	365	41	6300	700	0.971	0.958	1.47
	0 : 10	406	0	7000	0	0.971	-	-
ResNet-152 ArcFace	0 : 10	0	406	0	7000	-	0.920	-
	1 : 9	41	365	700	6300	0.974	0.920	3.09
	2 : 8	81	325	1400	5600	0.971	0.921	2.72
	3 : 7	122	284	2100	4900	0.968	0.922	2.42
	4 : 6	162	244	2800	4200	0.966	0.928	2.12
	5 : 5	203	203	3500	3500	0.963	0.928	1.96
	6 : 4	244	162	4200	2800	0.962	0.933	1.76
	7 : 3	284	122	4900	2100	0.961	0.936	1.65
	8 : 2	325	81	5600	1400	0.961	0.944	1.43
	1 : 9	365	41	6300	700	0.959	0.950	1.20
	0 : 10	406	0	7000	0	0.958	-	-

Table 5.5: **Test Set Img Imbalance.** The female and male accuracy for models trained on default train set computed on test set with various ratios of number of images per male and female identities. See details of the experiment in Section 5.5.2

Model	Img Ratio	# M ids	# F ids	# M imgs	# F imgs	M Acc	F Acc	Error Ratio
MFN CosFace	2 : 8	406	406	1400	5600	0.941	0.957	0.72
	3 : 7	406	406	2100	4900	0.959	0.956	1.06
	4 : 6	406	406	2800	4200	0.962	0.952	1.27
	5 : 5	406	406	3500	3500	0.967	0.946	1.64
	6 : 4	406	406	4200	2800	0.970	0.940	2.01
	7 : 3	406	406	4900	2100	0.973	0.925	2.75
	8 : 2	406	406	5600	1400	0.975	0.894	4.23
MFN ArcFace	2 : 8	406	406	1400	5600	0.939	0.956	0.72
	3 : 7	406	406	2100	4900	0.956	0.954	1.03
	4 : 6	406	406	2800	4200	0.961	0.951	1.26
	5 : 5	406	406	3500	3500	0.966	0.947	1.54
	6 : 4	406	406	4200	2800	0.969	0.941	1.91
	7 : 3	406	406	4900	2100	0.972	0.928	2.58
	8 : 2	406	406	5600	1400	0.974	0.901	3.87
ResNet-152 CosFace	2 : 8	406	406	1400	5600	0.921	0.938	0.78
	3 : 7	406	406	2100	4900	0.946	0.934	1.21
	4 : 6	406	406	2800	4200	0.952	0.927	1.51
	5 : 5	406	406	3500	3500	0.958	0.921	1.89
	6 : 4	406	406	4200	2800	0.962	0.912	2.32
	7 : 3	406	406	4900	2100	0.965	0.894	3.01
	8 : 2	406	406	5600	1400	0.967	0.855	4.37
ResNet-152 ArcFace	2 : 8	406	406	1400	5600	0.888	0.912	0.79
	3 : 7	406	406	2100	4900	0.916	0.909	1.09
	4 : 6	406	406	2800	4200	0.930	0.901	1.42
	5 : 5	406	406	3500	3500	0.940	0.889	1.85
	6 : 4	406	406	4200	2800	0.946	0.878	2.27
	7 : 3	406	406	4900	2100	0.950	0.853	2.92
	8 : 2	406	406	5600	1400	0.954	0.798	4.38

Table 5.6: **Train & Test Set Id Imbalance.** The female and male accuracy for models trained and tested on data with the same ratios of male and female identities. See details of experiment in Section 5.6.

Model	Ids Ratio	M Acc	F Acc	Error Ratio
MFN CosFace	0 : 10	-	0.945	-
	1 : 9	0.963	0.943	1.54
	2 : 8	0.966	0.947	1.56
	3 : 7	0.964	0.943	1.57
	4 : 6	0.967	0.945	1.63
	5 : 5	0.965	0.943	1.63
	6 : 4	0.968	0.947	1.63
	7 : 3	0.968	0.946	1.66
	8 : 2	0.969	0.946	1.72
	1 : 9	0.971	0.951	1.68
	0 : 10	0.972	-	-
MFN ArcFace	0 : 10	-	0.945	-
	1 : 9	0.962	0.946	1.42
	2 : 8	0.962	0.947	1.42
	3 : 7	0.962	0.943	1.52
	4 : 6	0.961	0.945	1.41
	5 : 5	0.964	0.944	1.54
	6 : 4	0.968	0.944	1.72
	7 : 3	0.967	0.946	1.61
	8 : 2	0.969	0.947	1.71
	1 : 9	0.968	0.949	1.63
	0 : 10	0.969	-	-
ResNet-152 CosFace	0 : 10	-	0.901	-
	1 : 9	0.943	0.906	1.65
	2 : 8	0.947	0.907	1.75
	3 : 7	0.947	0.902	1.86
	4 : 6	0.946	0.907	1.70
	5 : 5	0.946	0.912	1.64
	6 : 4	0.952	0.916	1.73
	7 : 3	0.955	0.919	1.79
	8 : 2	0.956	0.925	1.69
	1 : 9	0.954	0.931	1.49
	0 : 10	0.956	-	-
ResNet-152 ArcFace	0 : 10	-	0.880	-
	1 : 9	0.925	0.874	1.67
	2 : 8	0.924	0.878	1.61
	3 : 7	0.926	0.877	1.67
	4 : 6	0.925	0.877	1.63
	5 : 5	0.928	0.882	1.64
	6 : 4	0.930	0.890	1.58
	7 : 3	0.938	0.893	1.73
	8 : 2	0.937	0.900	1.59
	1 : 9	0.936	0.906	1.46
	0 : 10	0.942	-	-

Table 5.7: **Train & Test Set Img Imbalance.** The female and male accuracy for models trained and tested on data with the same ratios of number of images per male and female identity. See details of experiment in Section 5.6.

Model	Img Ratio	M Acc	F Acc	Error Ratio
MFN CosFace	2 : 8	0.821	0.923	0.43
	3 : 7	0.901	0.922	0.78
	4 : 6	0.928	0.919	1.12
	5 : 5	0.942	0.906	1.62
	6 : 4	0.952	0.892	2.24
	7 : 3	0.951	0.848	3.12
	8 : 2	0.954	0.740	5.70
MFN ArcFace	2 : 8	0.854	0.932	0.46
	3 : 7	0.916	0.933	0.79
	4 : 6	0.937	0.927	1.16
	5 : 5	0.951	0.919	1.64
	6 : 4	0.955	0.907	2.09
	7 : 3	0.957	0.871	3.01
	8 : 2	0.958	0.779	5.31
ResNet-152 CosFace	2 : 8	0.657	0.859	0.41
	3 : 7	0.832	0.873	0.76
	4 : 6	0.879	0.866	1.11
	5 : 5	0.912	0.848	1.74
	6 : 4	0.916	0.798	2.41
	7 : 3	0.922	0.695	3.90
	8 : 2	0.922	0.483	6.66
ResNet-152 ArcFace	2 : 8	0.638	0.840	0.44
	3 : 7	0.817	0.859	0.77
	4 : 6	0.870	0.855	1.12
	5 : 5	0.899	0.832	1.66
	6 : 4	0.911	0.792	2.34
	7 : 3	0.917	0.708	3.51
	8 : 2	0.915	0.488	6.02

Chapter 6: Technical Challenges for Training Fair Neural Networks

Joint work with Vedant Nanda, Micah Goldblum, John P. Dickerson and Tom Goldstein.

As machine learning algorithms have been widely deployed across applications, many concerns have been raised over the fairness of their predictions, especially in high stakes settings (such as facial recognition and medical imaging). To respond to these concerns, the community has proposed and formalized various notions of fairness as well as methods for rectifying unfair behavior. While fairness constraints have been studied extensively for classical models, the effectiveness of methods for imposing fairness on deep neural networks is unclear. In this work, we observe that these large models overfit to fairness objectives, and produce a range of unintended and undesirable consequences. We conduct our experiments on both facial recognition and automated medical diagnosis datasets using state-of-the-art architectures.

6.1 Introduction

Machine learning systems are increasingly deployed in settings with major economic and social impacts. In such situations, differences in model behaviors between different social groups may result in disparities in how these groups are treated [Barocas and Selbst, 2016, Galhotra et al., 2017]. For this reason, it is crucial to understand the bias of machine learning systems, and

to develop tools that prevent algorithmic discrimination against protected groups.

Much effort has been devoted to understanding and correcting biases in classical machine learning models (*e.g.* SVMs, Logistic Regression *etc.*). Overfitting is not a pernicious issue for classical models, and so fairness constraints that are imposed at train time often generalize to (unseen) test-time data. For overparameterized neural networks – as is often the case with modern deep neural networks [Zhang et al., 2017b] – our tools for understanding and controlling model bias are far less effective, in large part because of the difficulties created by overfitting. At train time, neural networks interpolate the data and achieve perfect accuracy on all sub-groups, thus making it impossible to obtain meaningful measures of bias on training data. When constraints are imposed using a sequestered dataset, the network may still overfit to constraints. Furthermore, the extremely fluid decision boundaries of neural networks open the possibility for complex forms of *fairness gerrymandering* [Kearns et al., 2018, Lipton et al., 2018], in which the decision boundaries of the model are moved to achieve a fairness constraint, while at the same time creating unintended consequences for important sub-groups that were not explicitly considered at train time. Since deep neural networks perform so much better than their linear counterparts on a wide range of tasks, it is important that we better understand the fairness properties of these complex systems.

This work investigates whether currently available methods for training fair models are capable of controlling bias in Deep Neural Networks (DNNs). We find that train-time fairness interventions – including those that have been thoroughly tested for classical ML models – are not effective for DNNs. We further show that when these fairness interventions do seem to work, they often result in the undesired phenomenon of fairness gerrymandering. Our contributions are as follows:

- We empirically test a range of existing methods for imposing fairness using constraints and penalties during training of DNNs. While methods of this type have been widely used and rigorously studied in the under-fitting regime (*i.e.* SVMs and linear models), we show that they fail in the overparameterized regime.

- We find that in some cases, fairness surrogates that work well for classical models do not work well for DNNs. In particular, equality of losses does not necessarily translate to equality of metrics used to evaluate performance (*e.g.* area under the curve).

- We also observe that specialized constraints designed for facial recognition often appear to work on training data, but fail on holdout classes. Facial recognition systems present a unique case because they operate differently during inference than during training.

- We consider adversarial methods for learning "fair features". In addition to discussing theoretical problems with these approaches, we observe that these methods are not effective at achieving fair outcomes in practice.

- We observe that fairness gerrymandering can be particularly problematic for DNNs because of their highly flexible decision boundaries. That is, parity along one sensitive attribute (*e.g.* sex) comes at the cost of increased disparity along another sensitive attribute (*e.g.* age).

We acknowledge that fairness constraints are not a universal solution to fair ML [see, e.g., Chapter 3 of Barocas et al., 2019]. Indeed, ML algorithms are only a small part of the bigger automated decision making system, and making these algorithms "fair" is only solving a small

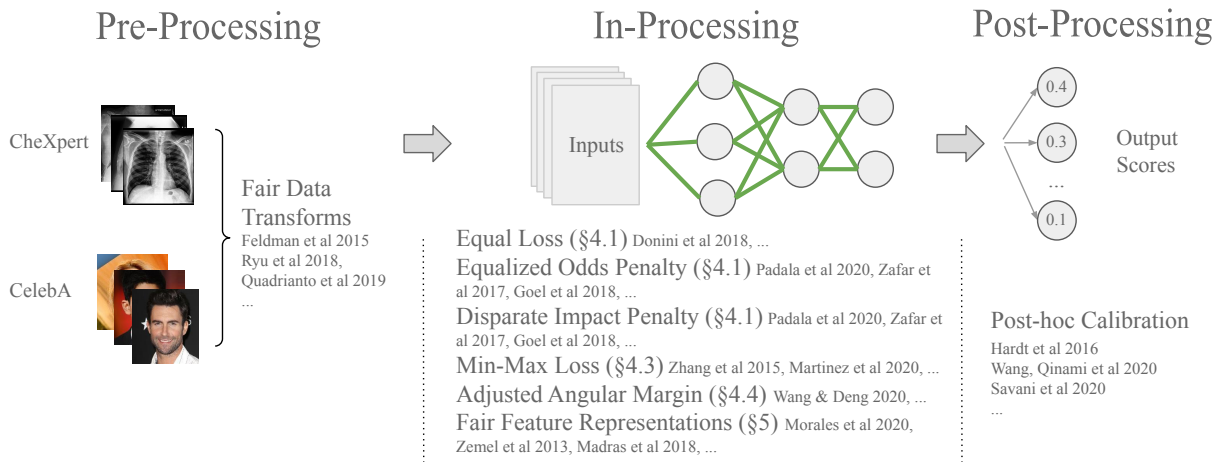

Figure 6.1: **[Brief Overview of Fairness in ML]** For the scope of this work, we consider only in-processing techniques and apply them to deep neural networks. We show that that over-parametrized nature of neural networks is one reason why current techniques fail.

part of what is a larger sociotechnical problem. However, it is important to understand the limitations of algorithmic fairness solutions. In our work we focus on analyzing the effectiveness of bias mitigation strategies in deep neural networks, and discuss the associated pitfalls.

6.2 Related work

There exist well documented cases of unfairness in key ML applications such as targeted ads [Speicher et al., 2018, Ali et al., 2019, Ribeiro et al., 2019], personalized recommendation systems [Biega et al., 2018, Singh and Joachims, 2018], credit scoring [Khandani et al., 2010], recidivism prediction [Chouldechova, 2017], hiring [Schumann et al., 2020], medical diagnosis [Larrazabal et al., 2020, Seyyed-Kalantari et al., 2020], facial recognition [Buolamwini and Gebru, 2018, Patrick J. Grother, 2010, Ngan and Grother, 2015], and others. This has resulted in a range of interdisciplinary work on understanding and mitigating bias in automated decision making systems [Binns, 2017, Leben, 2020, Hashimoto et al., 2018, Martinez et al., 2020, Nanda et al., 2021, Heidari et al., 2019]. Existing work on mitigating algorithmic unfairness can be

broadly put into three categories: pre-processing, in-processing and post-processing (see Figure 6.1). Works on pre-processing mostly focus on removing sensitive information from the data and building diverse and balanced datasets [Feldman et al., 2015, Ryu et al., 2018, Quadrianto et al., 2019, Wang and Deng, 2020]. In-processing aims to change the training routine, often via imposing constraints [Zafar et al., 2017a,b, 2019, Donini et al., 2018, Goel et al., 2018, Padala and Gujar, 2020, Agarwal et al., 2018, Wang and Deng, 2020] or by changing the optimization routine [Martinez et al., 2020, Diana et al., 2020, Lahoti et al., 2020]. Another in-processing strategy is to learn fair intermediate representations independent of sensitive attributes which lead to fairness on any downstream task [Dwork et al., 2012, Zemel et al., 2013, Edwards and Storkey, 2016, Madras et al., 2018, Beutel et al., 2017, Wang et al., 2019]. There isn't clear consensus on whether works along the lines of fair representation learning are pre-processing or in-processing (Zafar et al. [2019] categorize it as both in and pre). However, since most of these works rely on a learned model to perform the transformation of the dataset into a "fair" representation, for our purpose we consider these as in-processing. Post-processing techniques aim to change the inference mechanism to ensure fair outcomes [Hardt et al., 2016, Wang et al., 2020, Savani et al., 2020]. In this work, we limit our scope to understanding how in-processing techniques work for DNNs.

Our work is closely inspired by the works of Zafar et al. [2017a] that proposed smooth surrogates for various statistical notions of fairness which could then be imposed as constraints on training. However, their proposed method was only evaluated on linear models (such as logistic regression and SVMs) where an optimal solution can be efficiently found using constrained optimization. Padala and Gujar [2020] extended this line of work and proposed the use of Neural Networks to empirically estimate measures of fairness by considering class logits to be proxies

for class probabilities. They then applied these estimates of fairness as constraints to the training of neural networks using lagrangian multipliers. However, they only performed experiments on fully connected and shallow neural networks. We use their method of empirically estimating fairness measures and applying it as regularizers for training deep convolutional neural networks on image classification tasks. We find that their proposed approach does not give good fairness generalization in the highly overparametrized regime, such as the one we consider with our setup.

Most work on achieving fairness via in-processing methods in deep learning has been very tailored to a particular task like facial recognition [Wang and Deng, 2020] or achieves a very specific kind of fairness, such a removing sensitive attribute information from a representation [Morales et al., 2020]. In addition to these tailored approaches, we also apply the more general approach of enforcing various definitions of fairness via regularization during training, which until now had been primarily tested on either linear models, or very shallow deep neural networks.

Prior work has also suggested that algorithmic solutions for fairness are often hard to comprehend [Saha et al., 2020] or put into practice [Beutel et al., 2019]. Additionally, industry practitioners believe that fairness issues in real-world ML systems could be more effectively tackled by systematically changing the broader system design, rather than solely focusing on algorithmic "debiasing" [Holstein et al., 2019, Madaio et al., 2020]. Our work aims to show the fragility of algorithmic fairness interventions for deep models, thus extending the scholarship on challenges for fairness in ML. We refer readers to Barocas et al. [2019] for a broad and nuanced discussion of the abilities and limitations of fair machine learning.

6.3 Experimental setup

In this work, we investigate how different in-processing methods for mitigating algorithmic unfairness work on two different problems: facial recognition and medical image classification. We choose these two particular domains to illustrate broadly applicable pitfalls when applying train-time fairness interventions in deep learning. Moreover, while both facial recognition and medical image classification have seen unprecedented performance gains due to advances in deep learning, there have been well documented cases of bias and unfairness in both of these domains [Patrick J. Grother, 2010, Buolamwini and Gebru, 2018, Seyyed-Kalantari et al., 2020, Larrazabal et al., 2020]. Additionally, previous works have also outlined ethical and epistemic issues with facial recognition and algorithmic solutions to fairness in healthcare [McCradden et al., 2020, Raji et al., 2020, Andrejevic and Selwyn, 2020]. In this section, we describe our experimental setup which we use throughout the chapter.

6.3.1 Face recognition

State-of-the-art facial recognition (FR) systems contain deep neural networks that extract facial features from **probe images** (new photos whose subject is identified by FR) and compare the resulting features to those corresponding to **gallery images** (references with known identities). Probe images are matched to the gallery image whose features lie closest with respect to the cosine similarity.

We train facial recognition models with ResNet-18 and ResNet-152 backbones using the popular CosFace head, designed to increase angular margins between identities [Wang et al., 2018]. All of our models are trained as classifiers using focal loss [Lin et al., 2017]. We split the

Celeb-A dataset into train (9,177 identities) and test (1,000 identities) sets with non-overlapping identities. Therefore, at test time, the FR model is evaluated on people whose images it has not seen during training. We split a validation set from the training data consisting of 3 images from each identity with more than 6 images.

We measure the performance of FR systems with two methods. On training data we can use the classification head from training to report multi-label classification accuracy, while for validation and test sets we rip off the classification head and report **rank-1** nearest neighbor (in feature space) accuracy as is mainstream in the facial recognition literature. We find that standard facial recognition models exhibit lower testing accuracy for females than for males, see Table 6.3. This accuracy gap does not result from unbalanced train data; in fact female identities have more gallery images on average than male identities and 58% of identities in the data are female. Note that validation accuracy is lower than test accuracy in all of our experiments since the validation set contains 9 times as many classes and fewer images per class compared to the test set.

6.3.2 Medical image classification

We use CheXpert [Irvin et al., 2019], a widely used and publicly available benchmark dataset for chest X-ray classification. This dataset consists of 224,316 chest radiographs annotated with a binary label indicating the presence of a given pathology. We consider the following 5 pathologies: Cardiomegaly (CA), Edema (ED), Consolidation (CO), Atelectasis (AT) and Pleural Effusion (PE). This yields a *multi-label* classification task, predicting which of the 5 pathologies are present given a chest x-ray image. We train models using weighted binary cross entropy loss, and we report performance via area under the curve (AUC) for each of the 5 tasks.

Our experiments only use images for which sex and age labels are available, yielding a total of 223,413 images. We randomly split this data in a 80:20 ratio to form the training and validation sets respectively. The dataset provides a test set with 234 images labelled by radiologists. The training set is primarily composed of males (60%), while validation and test sets are more balanced (55% males).

We use the highest ranked model on the CheXpert leaderboard[1] with a publicly available implementation.[2] We fine-tune a Densenet121 [Huang et al., 2017] pre-trained on ImageNet [Russakovsky et al., 2015] for this task. Additional details about the model, data preprocessing, optimizer, and hyperparameters can be found in Section 6.9.

6.4 Fairness notions

We consider the traditional fair machine learning setup consisting of a training dataset $\mathcal{D} = \{(x_i, a_i, y_i)\}_{i=1}^N$, where x_i are drawn independently from the input distribution $\mathcal{X}$, $a_i \in \mathcal{A}$ are sensitive features (such as race, gender *etc.*) and $y_i \in \mathcal{Y}$ are true labels. For simplicity's sake, assume that $\mathcal{A} = \{0, 1\}$. $\mathcal{Y}$ is binary in the medical imaging task and multi-class for facial recognition. We wish to train a model $f_\theta : \mathcal{X} \to \mathbb{R}$ which can predict an outcome $\hat{y}_i$ for a given x_i. Standard training procedures outside of the fair training regime minimize the average loss over training samples, $\hat{L}(f_\theta)$. We refer to this as the *baseline* training scheme in our experiments. We refer to average loss on data points where $a_i = 1$ as $\hat{L}^{a+}(f_\theta)$ and points where $a_i = 0$ as $\hat{L}^{a-}(f_\theta)$. Below, we recall fairness notions which we use in this work and also describe how we operationalize them.

[1] https://stanfordmlgroup.github.io/competitions/chexpert/
[2] https://github.com/jfhealthcare/Chexpert

Accuracy equality requires the classification system to have equal misclassification rates across sensitive groups [Zafar et al., 2017a,b, 2019].

$$\mathbb{P}(\hat{y} \neq y | a = 0) \approx \mathbb{P}(\hat{y} \neq y | a = 1) \tag{6.1}$$

Because accuracy is a discontinuous function of the model parameters, we use equal loss as a surrogate, and solve:

$$\min_{\theta} \left[\hat{L}(f_\theta) + \alpha \left| \hat{L}^{a+}(f_\theta) - \hat{L}^{a-}(f_\theta) \right| \right] \tag{6.2}$$

Equalized odds aims to equalize the true positive and false positive rates for a classifier (sometimes also referred to as *disparate mistreatment*) [Hardt et al., 2016].

$$\mathbb{P}(\hat{y} = 1 | a = 1, y = y) \approx \mathbb{P}(\hat{y} = 1 | a = 0, y = y) \tag{6.3}$$

The Equalized Odds Penalty [Padala and Gujar, 2020] aims to approximate equalized odds using logits as measures of probability. Thus, we minimize the following objective:

$$\min_{\theta} \left[\hat{L}(f_\theta) + \alpha(fpr + fnr) \right], \text{ where} \tag{6.4}$$

$$fpr = \left| \frac{\sum_i p_i(1 - y_i)a_i}{\sum_i a_i} - \frac{\sum_i p_i(1 - y_i)(1 - a_i)}{\sum_i (1 - a_i)} \right| \tag{6.5}$$

$$fnr = \left| \frac{\sum_i (1 - p_i)y_i a_i}{\sum_i a_i} - \frac{\sum_i (1 - p_i)y_i(1 - a_i)}{\sum_i (1 - a_i)} \right| \tag{6.6}$$

Here, p_i denotes a softmax output (binary prediction task).

Disparate impact is a widely adopted notion that requires any decision making process'

outcomes to be independent of membership in a sensitive group [Calders et al., 2009, Barocas and Selbst, 2016, Chouldechova, 2017, Feldman et al., 2015]:

$$\mathbb{P}(\hat{y} = 1|a = 1) \approx \mathbb{P}(\hat{y} = 1|a = 0) \tag{6.7}$$

The Disparate Impact Penalty [Padala and Gujar, 2020] aims to approximate disparate impact through the objective,

$$\min_{\theta} \left[\hat{L}(f_\theta) + \alpha di \right], \text{ where} \tag{6.8}$$

$$di = -\min \left(\frac{\sum_i a_i p_i / \sum_i a_i}{\sum_i (1 - a_i) p_i / \sum_i (1 - a_i)}, \frac{\sum_i (1 - a_i) p_i / \sum_i (1 - a_i)}{\sum_i a_i p_i / \sum_i a_i} \right) \tag{6.9}$$

Max-Min fairness focuses on maximizing the performance for the most discriminated against group, i.e. the group with lowest utility [Rawls, 1971, Zhang and Shah, 2014, Hashimoto et al., 2018, Mohri et al., 2019, Martinez et al., 2020, Diana et al., 2020, Lahoti et al., 2020].

$$\max \min_{a \in \mathcal{A}} \mathbb{P}(\hat{y} = y|a) \tag{6.10}$$

To optimize models for Max-Min fairness, we minimize the loss for the sensitive group with maximum loss at the current iteration. That is, we perform the following optimization at each iteration:

$$\min_{\theta} \ \max \left\{ \hat{L}^{a+}(f_\theta), \hat{L}^{a-}(f_\theta) \right\} \tag{6.11}$$

Since both equality of opportunity and disparate impact notions assume existence of a beneficial outcome, they are most useful for binary classification tasks, and we only use them in the medical

image classification task.

6.4.1 Training with fairness regularization

One common approach for mitigating unfairness is through imposing fairness constraints or regularizers on the training objective. In this section, we describe the effectiveness of various regularization-based methods at improving the fairness of models trained for medical image classification and facial recognition.

CheXpert We implement three types of regularizers: equal loss, disparate impact, and equality of opportunity penalties, with the aim to achieve parity in performance (*i.e.* AUC scores) for both males and females. In all previous works that apply such constraints, the experiments are either performed on linear models (*e.g.* SVM in [Donini et al., 2018]) or on small neural networks (*e.g.* 2-layer network in [Padala and Gujar, 2020]). Under such settings, it is reasonable to assume that one can reliably measure fairness notions on the train set and expect such fairness to generalize to an unseen test set. However, as we see in our experiments, this is seldom the case with DNNs, which are highly overparametrized and can easily fit the train data [Zhang et al., 2017b]. Hence, in theory, these regularizers will be ineffective since the regularizer's value will be extremely low on the train set purely as a result of overfitting.

We observe a similar trend in our empirical results reported in Table 6.1[3]. The model performs very well on the train set and thus appears to be fair, where fairness is measured by the difference in AUC value of males and females. However, when evaluated on the test set, *models trained with the regularizers can be even less fair than a baseline model*. There is one noticeable exception in Table 6.1; the equal loss regularizer is able to achieve better parity between AUC

[3]Our full slate of results can be found in Section 6.10.

Table 6.1: **[CheXpert - training with fairness penalties]** Results for all 5 CheXpert tasks: Cardiomegaly (CA), Edema (ED), Consolidation (CO), Atelectasis (AT) and Pleural Effusion (PE). The regularizer is optimized on the training data, and α here denotes the coefficient of the regularizer.

Scheme	Task	Train				Test			
		M	F	Gap	Penalty	M	F	Gap	Penalty
Baseline	CD	0.905	0.902	0.004	0.006	0.712	0.691	0.046	0.016
	ED	0.857	0.844	0.013	0.015	0.905	0.849	0.057	0.022
	CO	0.832	0.834	0.004	0.004	0.824	0.760	0.069	0.163
	AT	0.716	0.709	0.008	0.002	0.804	0.756	0.065	0.004
	PE	0.873	0.880	0.007	0.013	0.851	0.923	0.072	0.433
	Avg	**0.837**	**0.834**	**0.007**	**0.008**	**0.819**	**0.796**	**0.062**	**0.127**
Eq. Loss Train $\alpha = 100$	CD	0.934	0.932	0.002	0.005	0.764	0.758	0.005	0.001
	ED	0.879	0.872	0.007	0.016	0.914	0.875	0.039	0.019
	CO	0.888	0.888	0.003	0.003	0.840	0.849	0.026	0.136
	AT	0.732	0.727	0.005	0.002	0.825	0.772	0.053	0.003
	PE	0.882	0.886	0.004	0.005	0.861	0.897	0.036	0.387
	Avg	**0.863**	**0.861**	**0.004**	**0.006**	**0.841**	**0.830**	**0.032**	**0.109**
Eq. Loss Train $\alpha = 1000$	CD	0.820	0.814	0.007	0.001	0.805	0.770	0.035	0.002
	ED	0.816	0.799	0.018	0.004	0.888	0.811	0.077	0.010
	CO	0.680	0.686	0.006	0.000	0.909	0.800	0.109	0.010
	AT	0.625	0.613	0.012	0.002	0.740	0.719	0.021	0.001
	PE	0.843	0.851	0.009	0.002	0.834	0.924	0.090	0.248
	Avg	**0.757**	**0.753**	**0.010**	**0.002**	**0.835**	**0.805**	**0.066**	**0.054**
Disp. Impact Penalty $\alpha = 100$	CD	0.925	0.921	0.005	−0.211	0.759	0.743	0.028	−0.206
	ED	0.882	0.872	0.010	−0.316	0.921	0.843	0.077	−0.305
	CO	0.865	0.867	0.003	−0.181	0.799	0.730	0.074	−0.161
	AT	0.753	0.744	0.011	−0.526	0.766	0.690	0.076	−0.510
	PE	0.891	0.898	0.006	−0.839	0.878	0.907	0.030	−0.781
	Avg	**0.863**	**0.860**	**0.007**	**-0.415**	**0.825**	**0.783**	**0.057**	**-0.393**
Disp. Impact Penalty $\alpha = 1000$	CD	0.922	0.919	0.004	−0.196	0.795	0.725	0.070	−0.184
	ED	0.877	0.864	0.013	−0.381	0.901	0.848	0.053	−0.367
	CO	0.834	0.837	0.003	−0.213	0.705	0.673	0.040	−0.206
	AT	0.744	0.733	0.012	−0.532	0.818	0.734	0.085	−0.532
	PE	0.885	0.889	0.005	−0.834	0.846	0.903	0.057	−0.806
	Avg	**0.852**	**0.848**	**0.007**	**-0.431**	**0.813**	**0.777**	**0.061**	**-0.419**
Eq. Odds Penalty $\alpha = 100$	CD	0.934	0.932	0.004	0.009	0.743	0.745	0.002	0.005
	ED	0.889	0.880	0.008	0.024	0.883	0.836	0.047	0.019
	CO	0.879	0.884	0.004	0.002	0.776	0.790	0.045	0.083
	AT	0.771	0.768	0.008	0.005	0.769	0.699	0.070	0.010
	PE	0.895	0.900	0.005	0.005	0.840	0.893	0.053	0.185
	Avg	**0.874**	**0.873**	**0.006**	**0.009**	**0.802**	**0.792**	**0.043**	**0.060**
Eq. Odds Penalty $\alpha = 1000$	CD	0.929	0.924	0.005	0.006	0.774	0.768	0.021	0.004
	ED	0.896	0.894	0.005	0.022	0.893	0.872	0.022	0.023
	CO	0.866	0.867	0.003	0.003	0.777	0.742	0.035	0.088
	AT	0.760	0.757	0.003	0.003	0.754	0.761	0.038	0.013
	PE	0.895	0.901	0.006	0.006	0.870	0.940	0.070	0.175
	Avg	**0.869**	**0.869**	**0.005**	**0.008**	**0.814**	**0.817**	**0.037**	**0.061**

of males and females on the test set. However, we observe that this parity comes at the cost of increased disparity amongst age groups. This phenomenon is called *fairness gerrymandering*, which we discuss in detail in Section 6.7. All our models are trained using standard techniques to avoid overfitting (dropout and weight decay, more details in Section 6.9).

Thus, we conclude that achieving fairness via imposing constraints on the training set is challenging for DNNs. Their overparametrized nature leads to overfitting on training data and thus preventing any generalization of fairness on the test set. Moreover, overparametrization leads to a fluid decision boundary, which is prone to fairness gerrymandering.

Face recognition We find that applying an equal loss penalty on the difference in losses, even with a small coefficient, leads to improved fairness on the train set, although with a large accuracy trade-off. In many cases, training accuracy on females even becomes higher than training accuracy on males. At the same time, the validation and test accuracy do not decrease significantly, and the accuracy gap remains close to the gap of the baseline model. Increasing the penalty size leads to a higher accuracy trade-off, yet this still does not nearly eliminate the bias on the validation and test sets. One possible explanation for such behavior is that the model overfits on the fairness objective to its training data. Additionally, since the validation and testing behavior of FR systems involves discarding the classification head and only using the feature extractor, one might guess that fairness on training data was embedded in the classification head but not the feature extractor. Thus, equality of losses might be a good proxy for equality of accuracies computed using the classification head but not for accuracies computed using k-nearest neighbors in feature space on validation and test data. To test this hypothesis, we additionally compute classification accuracy on the validation set and find that fairness is not appreciably improved even in this case, so we conclude that the problem is of the overfitting nature. The detailed results can be found in

Table 6.2: **[Face Recognition ResNet-18]** Performance of models trained with different training schemes designed for mitigating disparity in misclassification rates between males and females. All models have ResNet-18 backbone and CosFace head. The first column refers to the training scheme used, penalty indicates the size penalty coefficient. The train accuracy is computed in a classification manner, while validation and test accuracies are computed in the 1-nearest neighbors sense. The gap subcolumn refers to the difference between male and female accuracies.

Scheme	Penalty	Train			Validation			Test		
		Male	Female	Gap	Male	Female	Gap	Male	Female	Gap
Eq. Loss Train	baseline	97.6	94.4	3.2	82.4	74.6	7.9	94.8	92.9	1.9
	$\alpha = 0.5$	72.7	75.1	-2.4	79.3	73.4	5.9	95	93.5	1.5
	$\alpha = 1$	28.2	29.5	-1.3	67.8	62	5.8	93	91.5	1.5
	$\alpha = 2$	0	0.1	-0.1	27	18.8	8.2	70.9	64.2	6.7
Eq. Loss Holdout	baseline	84	79.6	4.4	78.6	70.8	7.8	94.2	92.3	2.0
	$\alpha = 0.5$	59.3	46.9	12.4	75.2	66.5	8.7	94.5	92.5	2.0
	$\alpha = 1$	20.6	13	7.5	57.6	47.3	10.3	89.8	85.9	3.9
Min-Max	baseline	97.6	94.4	3.2	82.4	74.6	7.9	94.8	92.9	1.9
	fair	74.3	75.9	-1.7	79.4	73.4	6	95.5	93.4	2.1
Random Labels Flipping	baseline	97.6	94.4	3.2	82.4	74.6	7.9	94.8	92.9	1.9
	$p = 0.1$	92.4	91.8	0.6	81.3	75	6.3	95.4	93.2	2.1
	$p = 0.3$	68.2	86.6	-18.4	75.5	74.9	0.6	94.6	93.5	1.1
	$p = 0.5$	21.6	86	-64.4	67.3	73.3	-6	92.2	93	-0.7

Table 6.2 and in Section 6.10.

6.4.2 Imposing fairness constraints on a holdout set

If the only reason that fairness constraints on the training set are ineffective were that training accuracy is so high that models appear fair regardless of their test-time behavior, then one might be able to bypass this problem by imposing the fairness penalty on a holdout set instead.

However, as we see in our results in Tables 6.1, 6.2 and Section 6.10, this approach also fails. For both medical image classification and facial recognition tasks, in all of the cases where fairness is imposed on a holdout set, the downstream fairness on the test set deteriorates. We posit that the model overfits the penalty on the validation set, which ultimately harms fairness on the test set.

6.4.3 Max-Min training

Face recognition For FR models, Min-Max training results in similar behavior as applying the equality of losses penalty with a small coefficient. In particular, the training losses across genders indeed converge to similar values, and the model becomes more accurate on female identities at train time, but this result does not transfer to test data. In fact, the disparity in misclassification rates on the validation set improves marginally, while the test set accuracy gap deteriorates. Therefore, we again encounter an overfitting problem with fairness being achieved on the train set but not on unseen data.

CheXpert Table 6.1 displays the results for Max-Min training. We observe that such a training procedure does not improve fairness on the test set.

6.4.4 Adjusted angular margins for face recognition

As we mention in 6.3.1, facial recognition systems are trained using heads which increase the angular separation between classes. One way to improve fairness of the model with respect to gender is by using different angular margins during training and therefore promoting better feature discrimination for the minority class. Wang and Deng [2020] applied this idea to learn a strategy for finding the optimal margins during training for coping with racial bias in face verification.

To test this approach we train models with increased angular margin for females. To evaluate effectiveness of this method we follow Wang and Deng [2020] and measure mean intra- and inter-class angles in addition to accuracies. The intra-class angle refers to the mean angle between the average feature vector of an identity and feature vectors of images of the same person.

Table 6.3: **[Face recognition]** Accuracy and intra- and inter-class angles measured for male and female images for ResNet-152 model trained with **adjusted angle margins**. The numbers are measured on validation and test sets. It can be seen that 'fair' models (trained with increased angle margin for females) improve fairness on validation set, but increase the accuracy gap on test set.

	Penalty	Acc M	Acc F	Acc Gap	Intra M	Intra F	Inter M	Inter F
Validation	baseline	88.1	81.4	6.8	33.9	36.2	68.5	68.2
	fair	85.7	85.2	0.5	34.6	28.5	68.2	70.8
Test	baseline	96.6	94.5	2.1	43.7	47.5	70.5	70.1
	fair	95.9	91	4.9	44.7	49.1	69.6	68.4

Inter-class angle refers to the minimal angle between the average feature vector of an identity and average feature vectors of other identities. Intuitively, we would expect that increasing angular margin for females would decrease the intra-class angle and increase the inter-class angle for female identities.

Our results show that a model trained with increased angular margin for females achieves better validation accuracy, and intra- and inter-class separation for females. In fact, the accuracy gap on validation data drops from 6.8% to 0.5% for ResNet-152 model. However, these results do not transfer to test data which consists of photos of new identities. Ultimately, the accuracy and angle metrics worsen for female groups leading to increased misclassification rates across genders, see Table 6.3. These results indicate that adjusting angular margins for mitigating unfairness leads to an overfitting problem since fairness improves only on identities that appear in the training set.

6.5 Fair feature representations do not yield fair model behavior

Another strategy for mitigating unfairness is through learning fair representations that are not correlated with sensitive attributes in the hope that a classifier built on top of 'fair features'

will yield fair predictions. A recent paper introduces SensitiveNets, a sensitive information removal network trained on top of a pre-trained feature extractor with an adversarial sensitive regularizer [Morales et al., 2020]. Intuitively, the method learns a projection of embeddings $\varphi(x)$ that minimizes the performance of a sensitive attribute classifier while maximizing the performance of a face recognition system.

We apply this adversarial approach by training a sensitive information removal network for minimizing the facial recognition loss while simultaneously maximizing the probability of predicting a fixed gender class for all images. At the same time, the discriminator is trained for predicting the gender from $\varphi(x)$. Therefore, this can be formulated as a two-players game where the discriminator aims to predict the gender from features $\varphi(x)$, while the sensitive information removal network aims to output gender-independent features $\varphi(x)$ and confuse the discriminator. We find that when a network is trained without adversarial regularization, the discriminator predicts gender with 97% accuracy on the test set. Adversarial regularization with a sufficient penalty decreases the performance of a gender predictor to random, meaning that the resulting features $\varphi(x)$ are gender-independent.

The results show that when fair features $\varphi(x)$ are obtained through manipulating male images, *e.g.* when the adversarial regularizer forces the discriminator to label all images as females, the accuracy gap reduces at the expense of male accuracy. At the same time, when the adversarial regularizer manipulate female images, the model's accuracy gap only increases due to a drop in female accuracy. All results can be found in Table 6.4 and Section 6.10. Thus, we conclude that adversarial training decreases accuracy of the FR system on images whose feature vectors were used in the regularizer, damaging performance on all other classes.

There are principled mathematical reasons why "fair features" are problematic. When the

Table 6.4: **[Face recognition]** Facial recognition and gender classification test accuracy for ResNet-152 model trained with a sensitive information removal network on top. Here, α denotes the magnitude of adversarial penalty and for sufficiently large α, the discriminator predicts a fixed gender for all images (in bold). Gender in the first column is the gender of images penalized during training.

	α	Male	Female	Gap	Sens Acc
Penalize Females	$\alpha = 0$	95.9	93.3	2.6	97.0
	$\alpha = 1$	95.7	92.5	3.1	78.2
	$\alpha = 2$	95.1	91.0	4.1	**38.7**
	$\alpha = 3$	94.2	88.7	5.5	38.7
Penalize Males	$\alpha = 0$	95.9	93.3	2.6	97.0
	$\alpha = 1$	95.4	93.2	2.2	90.5
	$\alpha = 2$	92.8	91.9	0.9	**61.3**
	$\alpha = 3$	90.2	90.1	0.1	61.3

dataset is imbalanced, like Celeb-A which is majority women, the equilibrium strategy of a discriminator with no useful information is to always predict "female." In this case, the feature extractor, which has the goal of fooling the discriminator, can only do so by distorting the features of men to appear female – a strategy that disproportionately hurts accuracy of the male group. Furthermore, in the hypothetical scenario where groups are balanced and the training process succeeds in creating features with no gender information, it becomes impossible to create a downstream classifier that assigns any label to men at a different rate than women. This is true because if such a classifier existed, it could be used to create a better-than-random gender classifier. In cases where the distribution of labels is different for men and women, this means that false positive and false negative rates must differ across genders.

6.6 A simple baseline for fairness: label flipping

Many of the methods for fairness that we test above sacrifice accuracy without any gains in fairness on the testing data. Sometimes, they sacrifice both fairness and accuracy. Label flipping

is one way to navigate the trade-off by adjusting the accuracy of individual groups [Chang et al., 2020]. This is done by randomly flipping labels in the training data of the subgroup with superior accuracy. One might suppose that flipping labels will simply hurt performance, however we find that on facial recognition, this simple method can actually remedy unfairness without harming performance.

Face recognition For the facial recognition task, our models achieve higher accuracy on male images than on female images. Therefore, to decrease the accuracy gap, we randomly flip labels of a portion of male images during training. We do this by swapping male identities only with other random male identities so that female accuracy is largely preserved. We try different proportions of flipped labels: $p = 0.1, 0.3, 0.5$. Surprisingly, flipping 30% of male labels increases female test accuracy by 0.6% thereby decreasing the gender gap from 2.1% to 1.6% for ResNet-152 model. Also, flipping half of the male labels only drops male test accuracy from 96.6% to 95.2% and results in a 0.3% accuracy gap on test data, see results in Section 6.10.

CheXpert For CheXpert, we randomly flip the true label for $p = 0.01, 0.05,$ and 0.1 of samples from a particular sensitive group in each iteration. In general, we observe very unstable trends in AUC scores and disparities when using label flipping. For example, we observe that flipping 1% of male samples during training results in a major drop in AUC for males, averaged over all tasks (0.816 to 0.746) and a minor increase in AUC for females (0.781 to 0.784) resulting in increased disparity. However, with 5% flipped samples, we see that male AUC drops to a similar value (0.816 to 0.716) as the female AUC (0.781 to 0.705), thus leading to reduced disparity. A similar trend is seen when male samples are flipped. AUC values of models trained on randomly flipped data are consistently lesser than the baseline model. We thus conclude that random flipping might not be the best solution to achieving fairness in this setting, since it yields unreliable

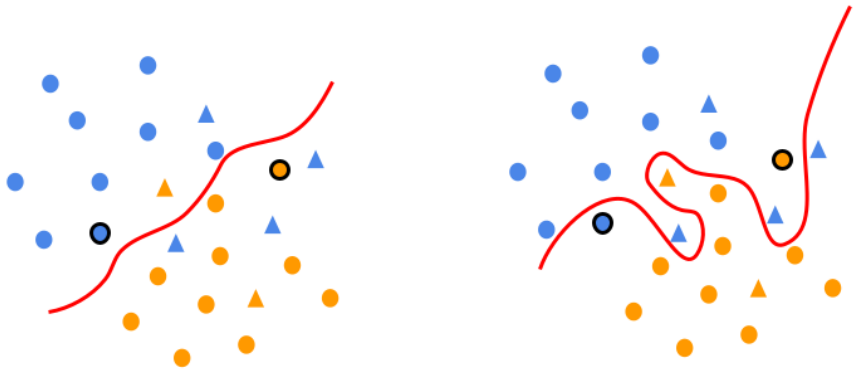

Figure 6.2: An illustration of gerrymandering; color denotes label, shape and outline are sensitive attributes. Model on the right is more fair to shape but less fair to outline.

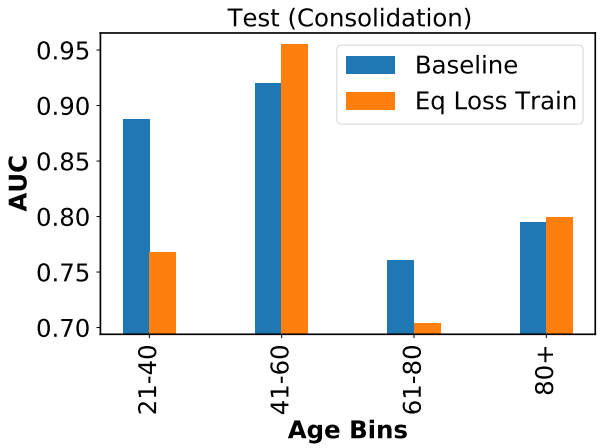

Figure 6.3: Gerrymandering behavior on CheXpert. The equal loss regularizer (in orange) achieves better parity along one demographic (sex, see Table 6.1 by making predictions more disparate along another demographic (age). For a model to be fair across age groups, the bars should all be of the same height.

trends in fairness and reliably performs worse than the baseline model. Results can be found in Section 6.10.

6.7 Fairness gerrymandering

Another unintended consequence of fairness interventions is fairness gerrymandering in which a model becomes more fair for one group but less fair to others [Kearns et al., 2018,

Table 6.5: **[Fairness gerrymandering on BUPT dataset]** The first row shows accuracies obtained with baseline model. The second, third, and fourth blocks reflect performance of models trained with equal loss penalty, adjusted angle margin for females, and randomly flipped labels for males respectively. For the models trained for "gender-fairness", we report differences from baseline.

Model	African	Asian	Caucasian	Indian
Baseline	92.8	90.4	93.7	94.5
Eq Loss	90.9	89.4	91.5	93.6
Diff	1.9	1.0	0.1	0.9
Margins	89.1	85.7	90.7	91.9
Diff	3.7	**4.7**	3	3.5
Flip	93.7	91.5	94	94.9
Diff	-0.9	**-1.1**	-0.3	-0.4

Lipton et al., 2018]. Similar/related images tend to clump together in feature space. For this reason, group-based fairness constraints that change the decision boundary are likely to induce label flips for entire groups of images with common features such as skin tone or age. Figure 6.3 (left) illustrates this phenomenon.

CheXpert In Table 6.1, we observed better parity for models trained with an equal loss penalty than baseline training. In this section, we take a closer look at how this affected disparities across another sensitive feature, age. Consolidation (CO) is the task for which disparity in AUC across males and females reduced the most. Figure 6.3 (right) shows that this reduction in disparity across males and females induced greater disparities across age groups, which is indicated by larger differences between subsequent age groups for the "fair" model. We see that fairness regularization with respect to gender leads to increased unfairness with respect to age.

Face recognition We investigate if face recognition models trained to be gender-fair suffer from gerrymandering. In particular, we consider models trained with the equal loss penalty, adjusted angle margins, and random label flipping. We evaluate our models on the race-labeled BUPT-Balancedface dataset [Wang and Deng, 2020] and find that models' performance changes dispro-

portionately with respect to race when trained for "gender fairness". For example, the FR model trained to have equal loss across genders is significantly less accurate on people of African origin than the baseline model, while the accuracy on Caucasian faces remains almost the same. The system trained with adjusted angle margins becomes even less fair to Asians, the group most discriminated against by the baseline model, with the accuracy gap between that on images of Asian and Indian people being increased by 1.2%, see Table 6.5). We then compare these results to a model trained with random label flipping. Surprisingly, the model trained on corrupted data improves accuracy for all four races, but the biggest improvement occurs on images of Asian individuals.

6.8 Discussion

We empirically demonstrate the challenges of applying current train time algorithmic fairness interventions to DNNs. Due to overparameterization, DNNs can easily overfit the training data and thus give a false sense of fairness during training which does not generalize to the test set. In fact, we observe that adding fairness constraints via existing methods can even exacerbate unfairness on the test set. In cases where train-time fairness interventions are effective on the test set, we observe fairness gerrymandering. We posit that overparameterization makes the decision boundary of the learned neural network extremely fluid, and changing it to conform to fairness along a certain attribute can hurt fairness along another sensitive attribute. Additionally, we observe that using a holdout set to optimize fairness measures also does not yield fair outcomes on the test set, due to both overfitting and bad approximation by fairness surrogates. Our results outline some of the limitations of current train time interventions for fairness in deep

learning. Evaluating other kinds of existing fairness interventions, such as pre-processing and post-processing for overparameterized models, as well as building better train time interventions are interesting avenues for future work.

6.9 Experimental Details and Additional Results

6.9.1 Medical Image Classification - CheXpert

Data Pre-Processing

CheXpert contains chest x-ray images of different patients from different angles and thus does not have a fixed resolution. We resize each image to 256x256. We use the same pre-processing pipeline found in `https://github.com/jfhealthcare/Chexpert`. We add a Gaussian blur with $\sigma = 3$ and mean normalize each image with a mean of 128.0 and standard deviation of 64.0.

Data Augmentation

We augment data by duplicating each image with a random affine transformation with rotations between -15 and 15 degrees, a vertical and horizontal translation of 0.05, scaling between 0.95 and 1.05. We fill the areas outside the transformed region with gray color (128 RGB value). Pre-processing steps are applied to each augmented image.

Training Details

We use a batch size of 56 for all our experiments. For some of the label flip experiments, we needed to increase the batch size to 200 so as to find a non-zero number of samples to flip. Each model was trained for 20 epochs. We used the Adam optimizer with a learning rate of 0.0001 and learning rate drops by a factor of 0.1 every epoch. This is the same as done by the authors of `https://github.com/jfhealthcare/Chexpert`. For the experiments where we added a fairness regularizer (Eq 6.2, 6.4, 6.8), we need to choose an additional hyperparameter α. We try a range of reasonable values and report results for the best α. We also show results

for other α values in Section 6.10, however we observe that higher values can interfere with the usual training objective.

6.9.2 Face Recognition

Training routine

We train facial recognition systems with ResNet-18 and ResNet-152 backbones and CosFace head using Focal Loss for 120 epochs with a batch size of 512. We utilize the SGD optimizer with a momentum of 0.9, weight decay of 5e-4 and learning rate of 0.1, and we drop the learning rate by a factor of 10 at epochs 35, 65 and 95. All images used for training contain aligned faces re-scaled to 112×112. During training we use random horizontal flip data augmentation.

For training routines, we modify the code from publicly available github repository face.evoLVe.PyTorch [4].

Training with fairness constraints

For facial recognition, we only apply an equal loss penalty, that is the absolute value of the difference between focal losses computed on male and female images in a batch. When regularization is imposed on training data, the same images are used to compute the classification loss and fairness penalty. When regularization is imposed on a holdout set, 10% of training images are kept for enforcing the fairness penalty and are not used in the recognition objective.

Adjusted Angular Margins

Below, we provide the loss for CosFace:

$$L_C = \frac{1}{N}\sum_{i=1}^{N} -\log \frac{e^{s(\cos(\theta_{y_i,i})-m)}}{e^{s(\cos(\theta_{y_i,i})-m)} + \sum_{i=1,i\neq y}^{n} e^{s\cos(\theta_{j,i})}},$$

[4] https://github.com/ZhaoJ9014/face.evoLVe.PyTorch

where x_i is the feature vector from the i-th sample from class $y^{(i)}$. W denotes the weight matrix of the last layer. Then, W_j is the j-th column of W and θ_{ij} denotes the angle between W_i and x_j. A fixed parameter $m \geq 0$ controls the magnitude of the cosine margin.

When trained regularly, $m = 0.35$ is used for both female and male images. When angular margin is adjusted for females, $m = 0.75$ is used for females and $m = 0.35$ for males.

Sensitive Information Removal Network

We denote the parameters of the sensitive information removal network (SIRN) as w and parameters of the sensitive classifier on top of it as w_s. SIRN takes as an input pre-trained embedding x_i of an image i from identity y_i and sensitive group s_i (gender) and outputs its projection $\varphi(x)$. The modified embedding is then fed into the CosFace head which outputs the logits for identities and into the sensitive head that outputs logits for genders. SIRN consist of 4 linear layers with ReLU-nonlinearities and sensitive head consists of 3 linear layers with ReLU-nonlinearities. Let L_{FR} denote the focal loss for the facial recognition task and L_S denote the cross-entropy loss for the sensitive attribute classification task.

Then, the optimization objective for the problem is

$$\min_{w} \frac{1}{N} \sum_{i} L_{FR}(\varphi(x_i), y_i) + \alpha \log(1 + |0.9 - P_s(s|\varphi(x_i))|)$$

$$\min_{w_s} \frac{1}{N} \sum_{i} L_S(\varphi(x_i), s_i),$$

where s is a fixed sensitive group (fixed gender), and α is the magnitude of the adversarial regularization term. $P_s(s|\varphi(x_i)$ denotes the sensitive head logit corresponding to gender s. Therefore, the first objective minimizes the facial recognition loss while simultaneously maxi-

mizing the probability of predicting a fixed gender class for all images. At the same time, the second objective minimizes the classification loss of the sensitive attribute classifier.

6.10 Additional Results

Face Recognition Tables 6.6 and 6.7 show results for Face Recognition tasks. We also report results for additional hyperparameter values here.

CheXpert Tables 6.8, 6.9, and 6.10 show results for all CheXpert tasks. We also report results for additional hyperparameter values here.

Table 6.6: **[Face Recognition ResNet-18]** Performance of models trained with different training schemes designed for mitigating disparity in misclassification rates between males and females. All models have ResNet-18 backbone and CosFace head. The first column refers to the training scheme used, penalty indicates the size penalty coefficient. The train accuracy is computed in a classification manner, while validation and test accuracies are computed in the 1-nearest neighbors sense. The gap subcolumn refers to the difference between male and female accuracies. For the Fair Features training scheme, "gender" refers to the subgroup of images used in adversarial regularization during training.

Scheme	Penalty	Train			Validation			Test		
		Male	Female	Gap	Male	Female	Gap	Male	Female	Gap
Eq. Loss penalty on train set	baseline	97.6	94.4	3.2	82.4	74.6	7.9	94.8	92.9	1.9
	$\alpha = 0.5$	72.7	75.1	-2.4	79.3	73.4	5.9	95	93.5	1.5
	$\alpha = 1$	28.2	29.5	-1.3	67.8	62	5.8	93	91.5	1.5
	$\alpha = 2$	0	0.1	-0.1	27	18.8	8.2	70.9	64.2	6.7
Eq. Loss penalty on holdout set	baseline	84	79.6	4.4	78.6	70.8	7.8	94.2	92.3	2.0
	$\alpha = 0.5$	59.3	46.9	12.4	75.2	66.5	8.7	94.5	92.5	2.0
	$\alpha = 1$	20.6	13	7.5	57.6	47.3	10.3	89.8	85.9	3.9
Random Labels Flipping	baseline	97.6	94.4	3.2	82.4	74.6	7.9	94.8	92.9	1.9
	$p = 0.1$	92.4	91.8	0.6	81.3	75	6.3	95.4	93.2	2.1
	$p = 0.3$	68.2	86.6	-18.4	75.5	74.9	0.6	94.6	93.5	1.1
	$p = 0.5$	21.6	86	-64.4	67.3	73.3	-6	92.2	93	-0.7
Adjusted Margins	baseline	97.6	94.4	3.2	82.4	74.6	7.9	94.8	92.9	1.9
	fair	94.2	90.2	3.9	81	77.6	3.3	94.4	91.9	2.6
Min-Max	baseline	97.6	94.4	3.2	82.4	74.6	7.9	94.8	92.9	1.9
	fair	74.3	75.9	-1.7	79.4	73.4	6	95.5	93.4	2.1
Fair Features (Females)	baseline	98.8	97	1.8	81.5	74.5	7	93.7	91.2	2.6
	$\alpha = 1$	98.6	96.8	1.9	81.4	73.9	7.5	93.7	90.9	2.8
	$\alpha = 2$	98.4	96.3	2.1	81.2	72.9	8.3	93.7	90.6	3.1
	$\alpha = 3$	96.2	93.2	3	79.9	70.5	9.4	93.3	88.9	4.4
Fair Features (Males)	baseline	98.8	97	1.8	81.5	74.5	7	93.7	91.2	2.6
	$\alpha = 1$	98.7	96.8	1.9	81.1	74.2	6.9	93.6	91.2	2.4
	$\alpha = 2$	98.7	96.8	1.9	81.1	74.2	6.9	93.6	91.2	2.4
	$\alpha = 3$	90.3	86.6	3.7	75.3	71.9	3.4	89.5	89.9	-0.3

Table 6.7: **[Face Recognition ResNet-152]** Performance of models trained with different training schemes designed for mitigating disparity in misclassification rates between males and females. All models have ResNet-152 backbone and CosFace head.

Scheme	Penalty	Train			Validation			Test		
		Male	Female	Gap	Male	Female	Gap	Male	Female	Gap
Eq. Loss penalty on train set	baseline	99.6	99	0.6	88.1	81.4	6.8	96.6	94.5	2.1
	$\alpha = 0.5$	84.7	87.4	-2.7	86.4	80.9	5.5	97.2	95.3	1.9
	$\alpha = 1$	38.8	40.6	-1.8	76.4	70.4	6	95.4	94.2	1.2
	$\alpha = 2$	0	0	0	24.8	17.4	7.4	69.1	61.3	7.8
Random Labels Flip	baseline	99.6	99	0.6	88.1	81.4	6.8	96.6	94.5	2.1
	$p = 0.3$	80.9	94.9	-13.9	83.8	81.3	2.4	96.8	95.1	1.6
	$p = 0.5$	31.5	93.2	-61.7	77.8	80.8	-3	95.2	94.9	0.3
Adjusted Margins	baseline	99.6	99	0.6	88.1	81.4	6.8	96.6	94.5	2.1
	fair	99.1	98.4	0.7	85.7	85.2	0.5	95.9	91	4.9
Fair Features (Females)	baseline	99.8	99.6	0.2	87.3	80.6	6.7	95.9	93.3	2.6
	$\alpha = 1$	99.7	99.5	0.2	86.9	79.5	7.4	95.7	92.5	3.1
	$\alpha = 2$	99.4	98.9	0.5	86	77.1	8.9	95.1	91	4.1
	$\alpha = 3$	96.8	96	0.8	85.2	74.1	11.1	94.2	88.7	5.5
Fair Features (Males)	baseline	99.8	99.6	0.2	87.3	80.6	6.7	95.9	93.3	2.6
	$\alpha = 1$	99.7	99.5	0.2	86.3	80.2	6.1	95.4	93.2	2.2
	$\alpha = 2$	97.8	97.3	0.6	82.5	78.4	4	92.8	91.9	0.9
	$\alpha = 3$	91.9	90.1	1.8	79.4	75.7	3.7	90.2	90.1	0.1

Table 6.8: **[CheXpert - fairness penalties on the validation set]** Results for all 5 CheXpert tasks: Cardiomegaly (CA), Edema (ED), Consolidation (CO), Atelectasis (AT) and Pleural Effusion (PE).

Scheme	Task	Train			Validation			Test		
		Male	Female	Gap	Male	Female	Gap	Male	Female	Gap
Baseline	CD	.988	.990	.002	.826	.818	.007	.766	.739	.027
	ED	.953	.952	.000	.780	.777	.002	.885	.862	.024
	CO	.996	.996	.000	.681	.687	.006	.896	.808	.088
	AT	.879	.883	.004	.637	.631	.007	.637	.570	.068
	PE	.936	.942	.006	.841	.854	.013	.895	.925	.030
	Avg	**.950**	**.953**	**.002**	**.753**	**.753**	**.007**	**.816**	**.781**	**.047**
Eq. Loss Val	CD	.985	.987	.002	.827	.824	.002	.745	.669	.076
	ED	.931	.930	.001	.746	.735	.011	.885	.877	.008
	CO	.979	.984	.005	.663	.682	.019	.865	.680	.185
	AT	.863	.868	.005	.626	.629	.002	.759	.763	.004
	PE	.926	.932	.006	.839	.856	.017	.897	.933	.036
	Avg	**.937**	**.940**	**.004**	**.740**	**.745**	**.010**	**.830**	**.784**	**.062**
Eq. Odds Val	CD	.994	.996	.002	.776	.774	.002	.637	.715	.078
	ED	.936	.934	.003	.729	.721	.009	.858	.699	.159
	CO	.995	.995	.000	.670	.678	.008	.797	.705	.093
	AT	.867	.867	.001	.608	.595	.013	.671	.565	.106
	PE	.945	.953	.008	.830	.845	.015	.896	.917	.021
	Avg	**.947**	**.949**	**.003**	**.723**	**.722**	**.009**	**.772**	**.720**	**.091**
Disp. Impact Val	CD	.997	.997	.000	.795	.788	.007	.691	.708	.018
	ED	.972	.973	.001	.768	.760	.008	.853	.891	.038
	CO	.995	.996	.000	.641	.635	.006	.712	.640	.072
	AT	.874	.879	.005	.633	.626	.007	.786	.668	.118
	PE	.945	.951	.006	.830	.843	.013	.897	.921	.024
	Avg	**.957**	**.959**	**.003**	**.733**	**.730**	**.008**	**.788**	**.765**	**.054**

Table 6.9: **[CheXpert - minmax training]** Results for all 5 CheXpert tasks: Cardiomegaly (CA), Edema (ED), Consolidation (CO), Atelectasis (AT) and Pleural Effusion (PE).

Scheme	Task	Train			Validation			Test		
		Male	Female	Gap	Male	Female	Gap	Male	Female	Gap
Baseline	CD	.988	.990	.002	.826	.818	.007	.766	.739	.027
	ED	.953	.952	.000	.780	.777	.002	.885	.862	.024
	CO	.996	.996	.000	.681	.687	.006	.896	.808	.088
	AT	.879	.883	.004	.637	.631	.007	.637	.570	.068
	PE	.936	.942	.006	.841	.854	.013	.895	.925	.030
	Avg	**.950**	**.953**	**.002**	**.753**	**.753**	**.007**	**.816**	**.781**	**.047**
Min-Max Loss	CD	.992	.981	.011	.824	.820	.004	.771	.789	.018
	ED	.963	.904	.058	.774	.774	.000	.895	.838	.057
	CO	.988	.976	.012	.694	.713	.019	.886	.821	.065
	AT	.901	.771	.130	.657	.652	.004	.758	.826	.068
	PE	.946	.896	.050	.842	.852	.010	.891	.927	.036
	Avg	**.958**	**.906**	**.052**	**.758**	**.762**	**.008**	**.840**	**.840**	**.049**

Table 6.10: **[CheXpert - random label flipping]** Results for all 5 CheXpert tasks: Cardiomegaly (CA), Edema (ED), Consolidation (CO), Atelectasis (AT) and Pleural Effusion (PE).

Scheme	Task	Train			Validation			Test		
		Male	Female	Gap	Male	Female	Gap	Male	Female	Gap
Baseline	CD	.988	.990	.002	.826	.818	.007	.766	.739	.027
	ED	.953	.952	.000	.780	.777	.002	.885	.862	.024
	CO	.996	.996	.000	.681	.687	.006	.896	.808	.088
	AT	.879	.883	.004	.637	.631	.007	.637	.570	.068
	PE	.936	.942	.006	.841	.854	.013	.895	.925	.030
	Avg	**.950**	**.953**	**.002**	**.753**	**.753**	**.007**	**.816**	**.781**	**.047**
Random Flip $p = 0.1$	CD	.980	.943	.037	.814	.794	.020	.743	.790	.047
	ED	.930	.882	.048	.756	.750	.006	.892	.774	.118
	CO	.967	.897	.070	.656	.636	.020	.600	.667	.067
	AT	.853	.797	.056	.622	.594	.028	.668	.612	.056
	PE	.925	.914	.011	.841	.850	.009	.832	.916	.084
	Avg	**.931**	**.887**	**.044**	**.738**	**.725**	**.017**	**.747**	**.752**	**.074**
Random Flip $p = 0.05$	CD	.989	.961	.028	.790	.762	.028	.716	.634	.082
	ED	.950	.921	.030	.757	.744	.013	.889	.742	.147
	CO	.971	.941	.030	.684	.681	.003	.876	.758	.118
	AT	.854	.827	.027	.652	.632	.020	.689	.675	.014
	PE	.917	.910	.007	.829	.841	.013	.842	.906	.063
	Avg	**.936**	**.912**	**.024**	**.742**	**.732**	**.015**	**.802**	**.743**	**.085**
Random Flip $p = 0.01$	CD	.984	.981	.003	.786	.776	.010	.782	.683	.099
	ED	.949	.941	.008	.765	.756	.009	.856	.779	.076
	CO	.974	.972	.002	.647	.643	.003	.792	.794	.002
	AT	.821	.814	.007	.621	.608	.013	.770	.770	.000
	PE	.915	.917	.002	.817	.820	.003	.834	.916	.082
	Avg	**.929**	**.925**	**.004**	**.727**	**.720**	**.008**	**.807**	**.788**	**.052**

Bibliography

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

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.

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.

Woodrow Hartzog. The secretive company that might end privacy as we know it. *The New York Times*, Jan. 18 2020. URL `https://www.nytimes.com/2020/01/18/technology/clearview-privacy-facial-recognition.html`.

William Derringer. A surveillance net blankets china's cities, giving police vast powers. *The New York Times*, Dec. 17 2019. URL `https://www.nytimes.com/2019/12/17/technology/china-surveillance.html`.

Karen Weise and Natasha Singer. Amazon pauses police use of its facial recognition software. *The New York Times*, Jul. 10 2020. URL `https://www.nytimes.com/2020/06/10/technology/amazon-facial-recognition-backlash.html`.

Natasha Singer. Microsoft urges congress to regulate use of facial recognition. *The New York Times*, 2018.

Steve Lohr. Facial recognition is accurate, if you're a white guy. *New York Times*, 9, 2018.

Valeriia Cherepanova, Vedant Nanda, Micah Goldblum, John P Dickerson, and Tom Goldstein. Technical challenges for training fair neural networks. *arXiv preprint arXiv:2102.06764*, 2021.

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.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

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.

Ping-Yeh Chiang, Jonas Geiping, Micah Goldblum, Tom Goldstein, Renkun Ni, Steven Reich, and Ali Shafahi. Witchcraft: Efficient pgd attacks with random step size. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3747–3751. IEEE, 2020.

Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.

Zuxuan Wu, Ser-Nam Lim, Larry S Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 1–17. Springer, 2020.

Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 665–681. Springer, 2020.

Micah Goldblum, Avi Schwarzschild, Ankit Patel, and Tom Goldstein. Adversarial attacks on machine learning systems for high-frequency trading. In *Proceedings of the Second ACM International Conference on AI in Finance*, pages 1–9, 2021.

Emily Wenger, Josephine Passananti, Yuanshun Yao, Haitao Zheng, and Ben Y Zhao. Backdoor attacks on facial recognition in the physical world. *arXiv preprint arXiv:2006.14580*, 2020.

Yaoyao Zhong and Weihong Deng. Towards transferable adversarial attack against deep face recognition. *IEEE Transactions on Information Forensics and Security*, 16:1452–1466, 2020.

Daniel Pedraza, Dhaval Adjodah, Gretchen Greene, Josh Joseph, Thom Miano, and Francisco. Equalais. `https://equalais.media.mit.edu/`, 2018.

Parsa Saadatpanah, Ali Shafahi, and Tom Goldstein. Adversarial attacks on copyright detection systems. In *International Conference on Machine Learning*, pages 8307–8315. PMLR, 2020.

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.

Xiao Yang, Yinpeng Dong, Tianyu Pang, Hang Su, Jun Zhu, Yuefeng Chen, and Hui Xue. Towards face encryption by generating adversarial identity masks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3897–3907, 2021.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. In *ICLR*, 2021.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017a.

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.

Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision*, pages 87–102. Springer, 2016.

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.

Kai Zhang, Vítor Albiero, and Kevin W Bowyer. A method for curation of web-scraped face image datasets. In *2020 8th International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6. IEEE, 2020.

Ankan Bansal, Anirudh Nanduri, Carlos D Castillo, Rajeev Ranjan, and Rama Chellappa. Umdfaces: An annotated face dataset for training deep networks. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 464–473. IEEE, 2017.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

Jian Zhao. face.evolve: High-performance face recognition library based on pytorch. `https://github.com/ZhaoJ9014/face.evoLVe.PyTorch`, 2020.

Eli Schwartz, Raja Giryes, and Alex M. Bronstein. DeepISP: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing*, 28(2):912–923, Feb 2019. ISSN 1941-0042. doi: 10.1109/tip.2018.2872858. URL `http://dx.doi.org/10.1109/TIP.2018.2872858`.

Giulio Lovisotto, Simon Eberz, and Ivan Martinovic. Biometric backdoors: A poisoning attack against unsupervised template updating. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 184–197. IEEE, 2020.

Nicolas Papernot. A marauder's map of security and privacy in machine learning. *arXiv preprint arXiv:1811.01134*, 2018.

Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Mądry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1563–1580, 2022.

W Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. Metapoison: Practical general-purpose clean-label data poisoning. *Advances in Neural Information Processing Systems*, 33:12080–12091, 2020.

Jonas Geiping, Liam H Fowl, W. Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. Witches' brew: Industrial scale data poisoning via gradient matching. In *International Conference on Learning Representations*, 2021.

Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning-industry perspectives. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 69–75. IEEE, 2020.

Benjamin I.P. Rubinstein, Blaine Nelson, Ling Huang, Anthony D. Joseph, Shing-hon Lau, Satish Rao, Nina Taft, and J. D. Tygar. ANTIDOTE: Understanding and Defending Against Poisoning of Anomaly Detectors. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, IMC '09, pages 1–14, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-771-4. doi: 10.1145/1644893.1644895.

Neehar Peri, Neal Gupta, W. Ronny Huang, Liam Fowl, Chen Zhu, Soheil Feizi, Tom Goldstein, and John P. Dickerson. Deep k-NN defense against clean-label data poisoning attacks, 2019.

Sanghyun Hong, Varun Chandrasekaran, Yiğitcan Kaya, Tudor Dumitraş, and Nicolas Papernot. On the Effectiveness of Mitigating Data Poisoning Attacks with Gradient Shaping. *ArXiv200211497 Cs*, February 2020.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017a.

Chengyue Gong, Tongzheng Ren, Mao Ye, and Qiang Liu. Maxup: A simple way to improve generalization of neural network training. *arXiv preprint arXiv:2002.09024*, 2020.

Renkun Ni, Micah Goldblum, Amr Sharaf, Kezhi Kong, and Tom Goldstein. Data augmentation for meta-learning. In *International Conference on Machine Learning*, pages 8152–8161. PMLR, 2021.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6023–6032, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. *ArXiv170606083 Cs Stat*, June 2017b.

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. In *International Conference on Machine Learning*, pages 9389–9398. PMLR, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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*, CCS '16, pages 308–318, Vienna, Austria, October 2016. Association for Computing Machinery. ISBN 978-1-4503-4139-4. doi: 10.1145/2976749.2978318.

P Jonathon Phillips, Fang Jiang, Abhijit Narvekar, Julianne Ayyad, and Alice J O'Toole. An other-race effect for face recognition algorithms. *ACM Transactions on Applied Perception (TAP)*, 8(2):1–11, 2011.

Jacqueline G Cavazos, P Jonathon Phillips, Carlos D Castillo, and Alice J O'Toole. Accuracy comparison across face recognition algorithms: Where are we on measuring race bias? *IEEE transactions on biometrics, behavior, and identity science*, 3(1):101–111, 2020.

Mohsan Alvi, Andrew Zisserman, and Christoffer Nellåker. Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

Vitor Albiero, Krishnapriya KS, Kushal Vangara, Kai Zhang, Michael C King, and Kevin W Bowyer. Analysis of gender inequality in face recognition accuracy. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 81–89, 2020.

Brendan F Klare, Mark J Burge, Joshua C Klontz, Richard W Vorder Bruegge, and Anil K Jain. Face recognition performance: Role of demographic information. *IEEE Transactions on Information Forensics and Security*, 7(6):1789–1801, 2012.

Mei Wang and Weihong Deng. Mitigating bias in face recognition using skewness-aware reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9322–9331, 2020.

Matthew Gwilliam, Srinidhi Hegde, Lade Tinubu, and Alex Hanson. Rethinking common assumptions to mitigate racial bias in face recognition datasets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4123–4132, 2021.

Samuel Dooley, Ryan Downing, George Wei, Nathan Shankar, Bradon Thymes, Gudrun Thorkelsdottir, Tiye Kurtz-Miott, Rachel Mattson, Olufemi Obiwumi, Valeriia Cherepanova, et al. Comparing human and machine bias in face recognition. *arXiv preprint arXiv:2110.08396*, 2021.

Adam Kortylewski, Bernhard Egger, Andreas Schneider, Thomas Gerig, Andreas Morel-Forster, and Thomas Vetter. Empirically analyzing the effect of dataset biases on deep face recognition systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2093–2102, 2018.

P Jonathon Phillips, W Todd Scruggs, Alice J O'Toole, Patrick J Flynn, Kevin W Bowyer, Cathy L Schott, and Matthew Sharpe. Frvt 2006 and ice 2006 large-scale experimental results. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):831–846, 2009.

Joseph P Robinson, Gennady Livitz, Yann Henon, Can Qin, Yun Fu, and Samson Timoner. Face recognition: too bias, or not too bias? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–1, 2020.

Yaobin Zhang and Weihong Deng. Class-balanced training for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 824–825, 2020.

Patrick J Grother, Mei L Ngan, Kayee K Hanaoka, et al. Face recognition vendor test part 3: demographic effects. 2019.

Jason Van Hulse, Taghi M Khoshgoftaar, and Amri Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on Machine learning*, pages 935–942, 2007.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, 6(1):1–6, 2004.

Charles Elkan. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, volume 17, pages 973–978. Lawrence Erlbaum Associates Ltd, 2001.

Charles X Ling and Victor S Sheng. Cost-sensitive learning and the class imbalance problem. *Encyclopedia of machine learning*, 2011:231–235, 2008.

Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

Nitesh V Chawla, Aleksandar Lazarevic, Lawrence O Hall, and Kevin W Bowyer. Smoteboost: Improving prediction of the minority class in boosting. In *European conference on principles of data mining and knowledge discovery*, pages 107–119. Springer, 2003.

Yanmin Sun, Mohamed S Kamel, Andrew KC Wong, and Yang Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern recognition*, 40(12):3358–3378, 2007.

Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2): 539–550, 2008.

Paulina Hensman and David Masko. The impact of imbalanced training data for convolutional neural networks. *Degree Project in Computer Science, KTH Royal Institute of Technology*, 2015.

Hansang Lee, Minseok Park, and Junmo Kim. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In *2016 IEEE international conference on image processing (ICIP)*, pages 3713–3717. IEEE, 2016.

Samira Pouyanfar, Yudong Tao, Anup Mohan, Haiman Tian, Ahmed S Kaseb, Kent Gauen, Ryan Dailey, Sarah Aghajanzadeh, Yung-Hsiang Lu, Shu-Ching Chen, et al. Dynamic sampling in convolutional neural networks for imbalanced data classification. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, pages 112–117. IEEE, 2018.

Shoujin Wang, Wei Liu, Jia Wu, Longbing Cao, Qinxue Meng, and Paul J Kennedy. Training deep neural networks on imbalanced data sets. In *2016 international joint conference on neural networks (IJCNN)*, pages 4368–4374. IEEE, 2016.

Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587, 2017.

Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019.

David Danks and Alex John London. Algorithmic bias in autonomous systems. In *IJCAI*, volume 17, pages 4691–4697, 2017.

Samuel Dooley, George Z Wei, Tom Goldstein, and John P Dickerson. Robustness disparities in face detection. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Alexandra Chouldechova and Aaron Roth. A snapshot of the frontiers of fairness in machine learning. *Communications of the ACM*, 63(5):82–89, 2020.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6): 1–35, 2021.

Harini Suresh and John V Guttag. A framework for understanding unintended consequences of machine learning. *arXiv preprint arXiv:1901.10002*, 2, 2019.

Sheng Chen, Yang Liu, Xiang Gao, and Zhen Han. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In *Chinese Conference on Biometric Recognition*, pages 428–438. Springer, 2018.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

Christoph Dalitz. Reject options and confidence measures for knn classifiers. *Schriftenreihe des Fachbereichs Elektrotechnik und Informatik Hochschule Niederrhein*, 8:16–38, 2009.

Solon Barocas and Andrew D Selbst. Big data's disparate impact. *California Law Review*, 104: 671–732, 2016.

Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. Fairness testing: Testing software for discrimination. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, ESEC/FSE 2017, page 498–510, New York, NY, USA, 2017. doi: 10.1145/3106237.3106277. URL https://doi.org/10.1145/3106237.3106277.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017b.

Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International conference on machine learning*, pages 2564–2572. PMLR, 2018.

Zachary Lipton, Julian McAuley, and Alexandra Chouldechova. Does mitigating ml's impact disparity require treatment disparity? *Advances in neural information processing systems*, 31, 2018.

Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. http://www.fairmlbook.org.

Till Speicher, Muhammad Ali, Giridhari Venkatadri, Filipe Nunes Ribeiro, George Arvanitakis, Fabrício Benevenuto, Krishna P Gummadi, Patrick Loiseau, and Alan Mislove. Potential for discrimination in online targeted advertising. In *Conference on fairness, accountability and transparency*, pages 5–19. PMLR, 2018.

Muhammad Ali, Piotr Sapiezynski, Miranda Bogen, Aleksandra Korolova, Alan Mislove, and Aaron Rieke. Discrimination through optimization: How facebook's ad delivery can lead to biased outcomes. *Proceedings of the ACM on human-computer interaction*, 3(CSCW):1–30, 2019.

Filipe N Ribeiro, Koustuv Saha, Mahmoudreza Babaei, Lucas Henrique, Johnnatan Messias, Fabricio Benevenuto, Oana Goga, Krishna P Gummadi, and Elissa M Redmiles. On microtargeting socially divisive ads: A case study of russia-linked ad campaigns on facebook. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 140–149, 2019.

Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. Equity of attention: Amortizing individual fairness in rankings. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*, page 405–414, 2018.

Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2219–2228, 2018.

Amir E. Khandani, Adlar J. Kim, and Andrew W. Lo. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, 34(11):2767–2787, 2010.

Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163, 2017.

Candice Schumann, Jeffrey Foster, Nicholas Mattei, and John Dickerson. We need fairness and explainability in algorithmic hiring. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2020.

Agostina J Larrazabal, Nicolás Nieto, Victoria Peterson, Diego H Milone, and Enzo Ferrante. Gender imbalance in medical imaging datasets produces biased classifiers for computer-aided diagnosis. *Proceedings of the National Academy of Sciences*, 117(23):12592–12594, 2020.

Laleh Seyyed-Kalantari, Guanxiong Liu, Matthew McDermott, Irene Y Chen, and Marzyeh Ghassemi. Chexclusion: Fairness gaps in deep chest x-ray classifiers. In *BIOCOMPUTING 2021: proceedings of the Pacific symposium*, pages 232–243. World Scientific, 2020.

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.

P J. Phillips Patrick J. Grother, George W. Quinn. Report on the evaluation of 2d still-image face recognition algorithms. *NIST Interagency/Internal Report (NISTIR)*, 2010. URL `https://doi.org/10.6028/NIST.IR.7709`.

Mei L. Ngan and Patrick J. Grother. Face recognition vendor test (frvt) - performance of automated gender classification algorithms. *NIST Interagency/Internal Report (NISTIR)*, 2015. URL `https://doi.org/10.6028/NIST.IR.8052`.

Reuben Binns. Fairness in machine learning: Lessons from political philosophy. *Proceedings of Machine Learning Research*, 81:1–11, 2017.

Derek Leben. Normative principles for evaluating fairness in machine learning. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 86–92, 2020.

Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pages 1929–1938. PMLR, 2018.

Natalia Martinez, Martin Bertran, and Guillermo Sapiro. Minimax pareto fairness: A multi objective perspective. In *International Conference on Machine Learning*, pages 6755–6764. PMLR, 2020.

Vedant Nanda, Samuel Dooley, Sahil Singla, Soheil Feizi, and John P Dickerson. Fairness through robustness: Investigating robustness disparity in deep learning. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 466–477, 2021.

Hoda Heidari, Vedant Nanda, and Krishna Gummadi. On the long-term impact of algorithmic decision policies: Effort unfairness and feature segregation through social learning. In *International Conference on Machine Learning*, volume 97, pages 2692–2701, 2019.

Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

Hee Jung Ryu, Hartwig Adam, and Margaret Mitchell. Inclusivefacenet: Improving face attribute detection with race and gender diversity. *arXiv preprint arXiv:1712.00193*, 2018.

Novi Quadrianto, Viktoriia Sharmanska, and Oliver Thomas. Discovering fair representations in the data domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8227–8236, 2019.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial intelligence and statistics*, pages 962–970. PMLR, 2017a.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, pages 1171–1180, 2017b.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P Gummadi. Fairness constraints: A flexible approach for fair classification. *The Journal of Machine Learning Research*, 20(1):2737–2778, 2019.

Michele Donini, Luca Oneto, Shai Ben-David, John S Shawe-Taylor, and Massimiliano Pontil. Empirical risk minimization under fairness constraints. *Advances in neural information processing systems*, 31, 2018.

Naman Goel, Mohammad Yaghini, and Boi Faltings. Non-discriminatory machine learning through convex fairness criteria. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 116–116, 2018.

Manisha Padala and Sujit Gujar. Fnnc: Achieving fairness through neural networks. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence,{IJCAI-20}, International Joint Conferences on Artificial Intelligence Organization*, 2020.

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International conference on machine learning*, pages 60–69. PMLR, 2018.

Emily Diana, Wesley Gill, Michael Kearns, Krishnaram Kenthapadi, and Aaron Roth. Convergent algorithms for (relaxed) minimax fairness. *arXiv preprint arXiv:2011.03108*, 2020.

Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed Chi. Fairness without demographics through adversarially reweighted learning. *Advances in neural information processing systems*, 33:728–740, 2020.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013.

Harrison Edwards and Amos Storkey. Censoring representations with an adversary. In *International Conference in Learning Representations*, pages 1–14, 2016.

David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pages 3384–3393. PMLR, 2018.

Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075*, 2017.

Tianlu Wang, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5310–5319, 2019.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8919–8928, 2020.

Yash Savani, Colin White, and Naveen Sundar Govindarajulu. Post-hoc methods for debiasing neural networks. *arXiv preprint arXiv:2006.08564*, 2020.

Aythami Morales, Julian Fierrez, Ruben Vera-Rodriguez, and Ruben Tolosana. Sensitivenets: Learning agnostic representations with application to face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Debjani Saha, Candice Schumann, Duncan Mcelfresh, John Dickerson, Michelle Mazurek, and Michael Tschantz. Measuring non-expert comprehension of machine learning fairness metrics. In *International Conference on Machine Learning*, pages 8377–8387. PMLR, 2020.

Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Allison Woodruff, Christine Luu, Pierre Kreitmann, Jonathan Bischof, and Ed H Chi. Putting fairness principles into practice: Challenges, metrics, and improvements. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 453–459, 2019.

Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–16, 2019.

Michael A Madaio, Luke Stark, Jennifer Wortman Vaughan, and Hanna Wallach. Co-designing checklists to understand organizational challenges and opportunities around fairness in ai. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2020.

Melissa McCradden, Mjaye Mazwi, Shalmali Joshi, and James A Anderson. When your only tool is a hammer: ethical limitations of algorithmic fairness solutions in healthcare machine learning. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 109–109, 2020.

Inioluwa Deborah Raji, Timnit Gebru, Margaret Mitchell, Joy Buolamwini, Joonseok Lee, and Emily Denton. Saving face: Investigating the ethical concerns of facial recognition auditing. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 145–151, 2020.

Mark Andrejevic and Neil Selwyn. Facial recognition technology in schools: Critical questions and concerns. *Learning, Media and Technology*, 45(2):115–128, 2020.

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597, 2019.

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.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. Building classifiers with independency constraints. In *2009 IEEE international conference on data mining workshops*, pages 13–18. IEEE, 2009.

John Rawls. *A Theory of Justice*. Harvard University Press, 1971. ISBN 9780674000773. URL http://www.jstor.org/stable/j.ctvkjb25m.

Chongjie Zhang and Julie A Shah. Fairness in multi-agent sequential decision-making. *Advances in Neural Information Processing Systems*, 27, 2014.

Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pages 4615–4625. PMLR, 2019.

Hongyan Chang, Ta Duy Nguyen, Sasi Kumar Murakonda, Ehsan Kazemi, and Reza Shokri. On adversarial bias and the robustness of fair machine learning. *arXiv preprint arXiv:2006.08669*, 2020.