QUANTIFYING THE EXTENT TO WHICH POPULAR PRE-TRAINED CONVOLUTIONAL NEURAL NETWORKS IMPLICITLY LEARN HIGH-LEVEL PROTECTED ATTRIBUTES

Claudia Veronica Roberts

A MASTER'S THESIS

PRESENTED TO THE FACULTY

OF PRINCETON UNIVERSITY

IN CANDIDACY FOR THE DEGREE

OF MASTER OF SCIENCE IN ENGINEERING

RECOMMENDED FOR ACCEPTANCE BY

THE DEPARTMENT OF

COMPUTER SCIENCE

Adviser: Arvind Narayanan

June 2018

Abstract

In a popular technique called “transfer learning,” a technique used widely in the computer vision field, researchers adapt publicly released models pre-trained on millions of images, for example, to determine whether a person talking in a video is telling the truth. But could the resulting classifier be biased? Has the pre-trained neural network model learned high-level features that correspond to protected attributes such as race, gender, religion, or disability status? Understanding the high-level features encoded in deep neural network representations is pivotal to understanding the kinds of biases that may be introduced in a broad range of applications during transfer learning. In this paper, we quantify the extent to which three popular pre-trained convolutional neural networks are implicitly learning and encoding age, gender, and race information during the transfer learning process. Results indicate that these readily used pre-trained models encode information that can be used to infer protected attributes such as race, gender, or age, even with very limited labeled data available at a very high accuracy.

Acknowledgments

First and foremost, I would like to thank my adviser, Professor Arvind Narayanan, for his tutelage and guidance and for igniting my interest in this field of study. I would also like to thank Professor Olga Russakovsky for mentoring me and her valuable feedback.

Next, I would like to thank my co-authors Laura Roberts and Kenji Hata, both of whom I worked with closely to push out the first working iteration of this research project.

I would like to thank Professor Karl Ricanek Jr. of UNC Wilmington for providing eye coordinates for the MORPH dataset, and Ph.D. student Hachim El Khiyari of George Mason University for showing us how to extract face boundaries with the PhD face recognition MATLAB toolbox.

I would also like to thank Zhe Lin, Haoxiang Li, and Zhaowen Wang from Adobe Research for helpful discussions related to this research project, Professor Andrew Appel and Nicki Gotsis for expertly navigating me through the Master's program at Princeton University, Professor Thomas Funkhouser for encouraging me to seek out work I am passionate about and guiding me to Professor Arvind Narayanan, Professor William Massey and the Wesley L. Harris Scientific Society for creating a safe, inviting space for Princeton graduate engineering students to study each Sunday, and fellow Master's student Tosin Adewale for his encouragement and camaraderie.

Also, thank you to my friends, whom I appreciate so much. I feel so lucky to have y'all by my side as we travel together down this journey called life.

I want to give special thanks to my older sister, and co-author, Laura Roberts for always having my back, for inspiring me, for making me laugh, and for being an incredible role-model since day one. Following in your footsteps has never led me astray.

Also, special thanks to Stefan Wójcik for his never-failing support and for accompanying me on this adventure.

Finally, and most importantly, I would like to thank God for putting such wonderful people in my path and for giving me my parents. My parents have never tired from guiding me. I thank them for their wisdom, selflessness, and for all they've sacrificed to help me get to where I am today. They push me to dream big and be my best self, and give me the guidance, encouragement, and confidence to keep on pushing forward. Thank you, Daddy and Mama. I truly am, because of you.

1 Introduction

Artificial intelligence (AI) systems are increasingly influencing our interactions with the world and daily decision making. Today's AI systems rely on large amounts of data to detect patterns and learn generalized models on which to make predictions. Regrettably, these input data reflect the pervasive biases and historical prejudices of society. Thus, when AI systems are used in domains ranging from job candidate selection to image search, biased outputs are the norm rather than the exception. Left unchecked, these systems will perpetuate and amplify the impact of these biases on an unparalleled scale. Evidence suggests that this is a widespread and growing problem in modern day applications of AI technology [1, 18].

In the field of computer vision, these issues are further exacerbated by the inappropriate use of pre-trained models, image datasets that are not satisfactorily representative of minority groups, datasets that are a narrow representation of the world, and algorithmic learning models that act effectively like black-boxes. State-of-the-art CV systems are characterized by sophisticated deep learning algorithms, often involving multi-layer neural network architectures. The resulting models are opaque and not easily interpretable, in contrast to more interpretable models such as linear regression or decision trees. Moreover, companies that deploy such systems in their products are not keen to disclose their proprietary code and training sets. Thus, there are numerous instances of AI biases reported in the wild that have never been rigorously studied by the scientific community (e.g., Google Photos generation of offensive captions on images of black people [18].

One can conjecture that biases that arise as a result of training on images sourced directly from the web can be mitigated by generating responsibly curated datasets. However, this is (1) not practical at scale and (2) does not solve the problem. Curated computer vision image sets have been shown to be biased, both in the general sense of being inauthentic, closed representations of the visual world [60] and in the concrete sense of being biased with respect to gender roles and human actions [71]. More troubling, developers are using publicly available pre-trained, out-of-the-box neural network models in ways never anticipated by the researchers, resulting in complex pathways of bias propagation (e.g., using these models to detect a person's sexuality or whether they are a criminal simply by looking at his/her facial images [65, 63]). These factors make biases in computer vision especially important and challenging. Current literature has been motivated by domains such as credit scoring, job applicant candidacy, and criminal risk scoring, where issues of fairness and bias have traditionally been studied [9, 10]. Similar literature is lacking in the domain of computer vision for the aforementioned reasons. This has become even more apparent as government agencies across the globe move

towards using face recognition technology to track passengers who fly in and out of the country [49] and even track people as they go about their day [12]. In both cases, issues of fairness are inescapable. Thorough research and testing is required to understand how inaccuracies and biases in these facial recognition systems may be adversely affecting certain members of our society.

Our work seeks to (1) understand why and in what ways do dataset biases creep into trained deep learning models for computer vision and (2) find a way to combat this bias while preserving or improving performance on the original task. To that end, our first step is to gain an understanding of the high-level features encoded in deep neural network representations used in the computer vision field. To ensure fair outcomes, protected attributes (e.g., race, gender, religion, age, and sexual orientation) are typically withheld from human and algorithmic decision-makers. The emerging view is that this is not foolproof—algorithms and humans alike are still implicitly learning these protected attributes, producing biased and potentially unfair outcomes. Indeed, the success of deep learning models in computer vision can be largely attributed to their ability to learn features that capture high-level concepts from pixel data. In a popular technique called "transfer learning," users adapt publicly released models pre-trained on ImageNet [35], for example, to determine whether a person talking in a video is telling the truth. But could the resulting classifier be biased? Has the pre-trained neural network model learned high-level features that correspond to protected attributes such as race, gender, religion, or disability status? Understanding the high-level features encoded in deep neural network representations is pivotal to understanding the kinds of biases that may be introduced in a broad range of applications during transfer learning. Past research indicates that pre-trained CNNs do indeed implicitely learn high-level concepts such as object detectors and protected attributes such as race and gender [72, 34], but never before, to the best of our knowledge, have researchers formalized or quantified the extent to which these models are learning these things or measured their affect on tangential tasks during the transfer learning process.

The work in this paper encompasses the first phase in a larger research project goal. In this paper, we quantify the extent to which popular pre-trained computer vision deep learning models trained on ImageNet, namely VGG-16, ResNet-50, and Inception v3 [53, 24, 58], are implicitly learning and encoding features of interest during the transfer learning process. The features of interest are age, gender, and race, characteristics protected under the United States federal anti-discrimination law [61]. We run a series of experiments to determine the number of labeled image samples required to achieve a sufficiently acceptable level of predictive accuracy for each of the three attributes. A pre-trained convolutional neural network model that perfectly encodes the

protected attribute will only need a handful of labeled training samples. One that does not encode the desired feature will require thousands, if not millions, of samples to reach acceptable predictive accuracy.

We used the MORPH dataset for this project. The non-commercial release of the MORPH dataset contains over 55,000 unique images of more than 13,000 individuals ages 16 to 77 [47]. Each image is labeled with the individual's age on the day the picture was captured, and his/her race and gender. We divide the dataset into train, validation, and test sets. For each model type (i.e., VGG-16, ResNet-50, and Inception v3), we use transfer learning to train our models on the full training set and on five significantly smaller subsets of the train set. We use three transfer learning techniques: feature extraction, partial finetuning, and end-to-end finetuning. We are particularly interested in the test accuracy achieved by using what are considered to be in the computer vision community very small training sets. Results indicate that pretrained models encode information that can be used to infer protected attributes such as race, gender, or age, even with very limited labeled data available.

To have a basis for comparison, we run the same set of experiments using models pre-trained on VGGFace [42] and VGGFace2 [8], two facial image datasets that were created for the task of facial recognition. Since the MORPH dataset is itself a dataset of facial images, we expect the representations learned by models pre-trained on VGGFace and VGGFace2 to be specific to the visual structure of faces and thus, better suited for the attribute classification tasks, as opposed to the representations learned by models pre-trained on ImageNet, which learn the visual structure of the world.

We provide an overview of convolutional neural networks and transfer learning in Section 2. In Section 3, we provide a survey of the current fairness in machine learning and fairness in computer vision landscae and describe related work. To orient the user as to how this work fits into our larger research vision, we give an overview of our long-term research goals in Section 4. In Section 5 we detail our experimental setup and methods. We present and discuss our results in Section 6. Future work and extensions are considered in Section 7. We conclude in Section 8.

2 Technical Background

2.1 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is an artificial neural network that is specialized and optimized for taking in image pixel data as input [35]. Like a regular neural network, CNNs are composed of the following

important elements:

Score function A score function that maps input data to class scores. The simplest neural network, a 2-layer neural network with one hidden layer, might have the score function $s = W_2 \max(0, W_1 x)$ where x is the input data, W_1 and W_2 are a matrices of weights, and $\max(0, -)$ is a ReLU non-linearity that thresholds the input to 0. Other non-linearities (aka activation functions) include sigmoid and Tanh.

Objective function During the training process, the goal is to find an optimal set of weights W that produce class scores most congruent with the ground truth labels of the input training set. The function by which we measure how congruent the class scores are with the ground truth labels is called the loss function or objective function (or also the cost function). Thus, during training, the objective is to find W such that the loss function is minimized. A high loss means the computed class scores are incongruous. A low loss value means the computed class scores are in harmony with the ground truth labels of the input training data. There are various loss functions including cross-entropy loss for softmax classifiers and max-margin loss (aka the hinge loss) for SVM classifiers.

Optimization algorithm Finding a set of weights W that minimize the objective function is an optimization problem. One of the most popular optimization schemes starts off with randomly initialized weights and then uses gradient descent to iteratively update the weights in the negative direction of the gradient. The gradient tells us, along each dimension, the slope of the loss function or in other words, the direction in which the loss function has the steepest rate of increase (for our purposes of minimizing the loss we must go in the negative direction). The gradient of the objective function is computed via backpropagation, which is a recursive application of the calculus chain rule [50]. Once we have calculated the gradient of the loss function, we can perform one update of the weights along each feature dimension. We then once again calculate the class scores (the forward pass), calculate the loss, compute the gradients (the backward pass), update the parameters in the negative direction of the gradient, and repeat until we have reached the desired number of gradient descent iterations.

Regularization Often included in the objective function is a regularization penalty. The regularization loss controls how large we allow the weights in matrix W to grow and encodes into the total loss function

a preference for a certain set of weights over another. As long as the math checks out, it is possible to find multiple sets of weights that correctly classify every sample in a training set. However, we may prefer a set of weights that allows each input feature to have a small contribution to the final class score versus a set of weights that encourages a small subset of the input features to be highly influential to the final classifier. This preference scheme can be encoded into the loss function by using the $L2$ regularization penalty, for example.

Standard neural networks are composed entirely of fully-connected layers. Every neuron in a fully-connected layer is connected to every single other neuron in the previous layer. In effect, each neuron is performing a dot product with the input layer and its weights. This poses a problem when the input is images with large dimensions. Regular neural networks are not optimized for handling the explosive number of parameters created as a result. CNNs have a few additional differentiating elements that arise as a result of making the helpful assumption that x, the input data, are image pixels. These include:

Input data shape Whereas standard neural networks take in input data that is arranged in flat vectors or matrices and whose neurons in each layer are also in this arrangement, CNNs take in 3-dimensional input volumes.

Neuron connectivity Neurons in the convolutional layers of a CNN are connected to only a small region of the previous input layer, as opposed to all of the neurons. This greatly reduces the number of parameters compared to a fully connected network.

Parameter sharing A 2-dimensional cross-section of the 3-dimensional input volume is referred to as a depth slice [38]. Each depth slice, and as a result, each neuron in the depth slice, shares one set of learnable weights. This shared set of parameters is referred to as a filter or kernel.

Convolutional layer Convolutional layers consist of any number of filters. These small filters are then convolved across the input volume, i.e., we slide the filter across the entire width and height of the input volume computing the dot product between the weights in the filter and the current local input region. One pass of this filter across the input volume generates one 2-dimensional map of output values, called an activation map. There are as many activation maps as there are filters. Stacked together, these activation maps

create the output volume.

Other common layers Other common layers found in CNN architectures include the pooling layer, a layer which reduces the width and height of the input volume in order to reduce the number of parameters and the ReLU layer, which applies the $max(0, -)$ activation function element-wise on the passed in volume. CNNs are usually terminated by fully-connected layers, the layer which ultimately computes the final class scores.

2.2 Transfer Learning

Because training a CNN from scratch can take weeks to train and millions of labeled images, it is common for researchers in academia and industry alike to release their model architectures and the corresponding learned model weights. Many of the most popular and easily accessible models were originally trained on the ImageNet dataset [11], a dataset used for visual object recognition tasks. Using a technique called transfer learning, it is possible to leverage these pre-trained models for other visual classification tasks. The intuition behind why this is possible is that the lower layers of a CNN learn low level image features such as edges or color blobs. Regardless of your tasks, whether it is detecting cancerous tumors in CT scans or classifying whether a person in an image is sad, happy, or mad, your model will need to learn how to detect these generic features at some point. In fact, it is generally until you reach the higher layers of the neural network where the model starts learning the more training dataset specific features. Thus, in practice, it common to leverage these pre-trained models and tweak them as needed for the new visual task. Depending on how large the new dataset is and how similar the new visual recognition task is to the task the pre-trained model was originally trained to tackle, there are various degrees to which one can tweak the pre-trained model:

Feature Extraction This method manipulates the pre-trained model the least. Each image from the new dataset is passed through the model, and instead of using the final layer output, which contains the class scores for the original classification task, we use the penultimate layer output and treat the output as the new, transformed features of the image. These features, or CNN codes as they are sometimes called, are then used to train a simple linear SVM or softmax classifier.

Partial Fine-tuning This method involves freezing the first n layers of the pre-trained model and

then fine-tuning the upper portion of the neural network that contains details specific to the original task. In re-training the model with the new dataset by freezing some portion of the lower layers, you are tweaking the weights of the upper layers to be more useful for the new task while retaining the useful, more generic feature detectors at the bottom.

End-to-End Fine-tuning This method manipulates the pre-trained model the most by using the new image set to fine-tune all weights along the entire depth of the CNN.

3 Related Work

3.1 Fairness in Machine Learning

Machine learning is the science of creating algorithms that can automatically learn patterns from data. Given a set of input data, or "training data," these algorithms output a "model," or set of learned relationships, that can be used to make predictions on unseen data. These models can appear to be enigmatic black boxes to its down-stream users. Engineers of artificial intelligence systems have full discretion to choose the data and manipulate them and the model as they see fit. Moreover, companies that deploy such systems in their products are not keen to disclose their proprietary code and training datasets. In 2013, Latanya Sweeney discovered that Google AdSense was more likely to deliver advertisements for arrest records in response to queries of black-sounding names than to those of white-sounding names [57]. In this case, Sweeney confirmed that the differential ad-delivery seemed to be an artifact of Google's AdSense algorithm, which determines and ranks the advertisements shown to the user. In their investigation of COMPAS recidivism scores, Angwin et al. discovered that Northpointe's software was almost twice as likely to wrongly label black defendants as high risk as white defendants, and was more likely to mistake white defendants as low risk than black defendants [1]. The authors' investigation into whether Northpointe's software was more likely to predict a black person as having a higher likelihood of committing future crimes than a white person was made more difficult by a lack of insight into the algorithms that Northpointe was using to produce risk scores. As mentioned by Corbett et al., in some cases, neither the procedure nor the data used to generate these scores is disclosed, prompting worry that the scores are themselves discriminatory [10]. Since these two reports, interest in fairness in algorithmic learning has grown substantially. Researchers have responded by formalizing

various fairness definitions and criteria as well as developing the theoretical framework to prove that multiple fairness criteria cannot be achieved simulataneously [13, 9, 10, 29, 31, 44].

Researchers are actively exploring how to build classifiers that make decisions free from discrimination using adversarial approaches [15] and developing methods for reducing bias in learned classifiers by attempting to equalize false positive rates across individuals from different populations [3]. The mitigation of bias in machine learning is a research area that will continue to grow as policymakers, engineers, social scientist, educators, and users begin to feel the increased effect of algorithmic decision-makers in their lives.

3.2 Fairness in Computer Vision

Researchers have shown that various computer vision (CV) applications raise a multitude of privacy concerns, demonstrating that it is even possible to deduce a person's education level and median household income simply by looking at Google Street View images of cars [20]. Research stemming from a concern about CV applications with potential societal impacts seeks to improve the security and privacy of these applications [45, 55]. Only recently, however, have researchers started to focus on the societal impacts of biased CV models. This is in large part due to press articles signaling the emergence of police states and surveillance programs that seek to use face recognition in public spaces to identify and keep track of criminals and criminal activity as well as assit cross-country border screening through the use of biometric checkpoints [49, 12]. One study found that indeed, half of all American adults are currently in police face recognition databases [19]. Along with the privacy concerns this raises, issues of fairness, amplification of racial bias, and alarming false positive rates are also of much concern [19, 6].

Potential sources of unfairness in CV can be attributed to five things. 1) Lack of neural network interpretability. It has long been acknowledged by the machine learning and vision community that neural networks are blackboxes, which oftentimes lead to results that while satisfactory in terms of accuracy, are not easily understood or explained. Previous research efforts have sought to make neural networks more interpretable [69, 52, 68, 64, 40, 32, 2]. The hope is that with better model interpretability comes a better understanding of these models and ultimately, a better understanding of the various pathways of bias in neural network models. 2) Transfer learning in the CV is the norm not the exception. Transfer learning allows users to leverage pre-trained models and fine-tune them to their specific task. It continues to be used with great success [41, 66, 46]. But with great power comes great responsibility. Did the pre-trained models implicitly learn

high-level representations that are influencing the results of the newly learned model in an undesirable way unbeknownst to the user? This is the question we seek to answer in this paper and is the research goal of our longer term research projects. 3) Reliance on benchmark datasets and challenges. It is common in the CV community to evaluate the performance of their models against benchmark datasets such as the Imagenet large scale visual recognition challenge [51] and the MSR Image Recognition Challenge [22]. But when a research group achieves an error of 5% on one of these challenges, what does that mean? Learned on the general population, classifiers can exhibit marked differences in classification and prediction accuracy between members of majority groups and those of minority groups. Since the data has proportionally less information about minorities, the learned model learns statistical patterns that are only valid for the majority groups [23]. Given a model that does achieve 95% accuracy on a facial recognition task, how do we interpret the 5% error? Is it simply noise or is there a 5% error because we're 50% inaccurate on recognizing racial minority faces and 100% accurate on classifying the racial majority faces [23]? 4) Massive scale of recent datasets giving a semblance of fairness. Given the massive scale of these benchmark datasets, it may lead developers of algorithmic models to believe that their learned models are fair if trained and evaluated on these massive datasets. However, this has been shown not to be the case [30, 27, 5], and researchers continue to create datasets that are more diverse [48]. 5) Long-tail distributions in image datasets. Class imbalance in datasets not only leads to poor prediction accuracy on subjects from the minority group but also has been shown to be detrimental to the performance of convolutional neural networks in general [4]. The problem of long-tail distributions has lead researchers to explore techniques that use zero or few-shot learning [16, 17, 36].

3.2.1 Recognizing and Mitigating Bias in Neural Networks and Image Datasets

In the computer vision field, create image datasets might include scraping the web for image URLs and using Amazon Mechanical Turk (AMT) to identify objects in the images, as was done in the construction of the ImageNet dataset [35]. Or it could mean conducting a photo shoot of 50 different toys and changing the lighting placement for added variability [37]. While AMT has made it possible for researchers in computer vision to accomplish manual tasks like object detection at scale, researchers must contend with the unique challenges associated with crowdsourcing tasks to hundreds of "Turkers" all over the world. For example, researchers must identify and correct for workers trying to game or cheat the system. Computer vision researchers oftentimes find themselves developing methods to incentivize good Turkers and dissuade spam workers in addition to designing and developing the AMT project's user interface [33]. Training sets derived

from a process such as this one are oftentimes biased due to human manual labeling efforts. Asked to classify images of noise as sports balls or not, Indian Turkers were found to imagine red balls—the standard color for cricket balls and the most popular sport in India—and American workers seemed to imagine brown or orange balls–the standard color for basketballs and footballs [62]. Curated computer vision image sets have also been shown to be biased in a general sense in that they are inauthentic, closed representations of the visual world. Researches demonstrated that models trained on popular object recognition datasets had poor generalization ability when tested on other datasets with the same objects represented [60].

To mitigate biases that can occur due to disproportionate representation of minority groups in certain datasets, companies that build artifical intelligence systems can learn different models for different groups of people. However, this is not an easy task. While it is possible to learn two separate linear classifiers with equally acceptable predictive accuracies, an algorithm has yet to be discovered that can efficiently learn an arbitrary combination of the two linear classifiers [23]. The use of more fine-grained feature sets could be a potential solution to mitigating unintentionally discriminatory systems. This, however, is potentially expensive. The data collection process would be more expensive, it would be computationally more expensive, and adds undesired complexity to the system. Furthermore, it would force companies to look for more complex decision rules as well as place additional demands on engineers during the measurement and learning phase [23]. Other more feasible approaches involve directly manipulating the learned model. Classifical work on mitigating biases in neural networks include [28, 7, 70, 71] and [14].

4 Overall Research Vision

We propose that there are four major questions the computer vision community needs to address with regards to the responsible use of pre-trained models for tangentially related visual tasks. Together, these four questions comprise a larger corpus of research work that can subsequently be broken down into four individual research goals or steps in the overarching research vision.

1. Quantifying the extent to which neural networks are implicitly learning high-level concepts, specifically protected attributes.
2. Understanding how these implicitly learned high-level representations of protected attributes may be influencing the decisions made in new visual tasks during the transfer learning process and understanding the extent to which they are influencing the final decision.

3. If we find that they are indeed influencing the decisions to some non-trivial degree...understanding in what contexts this behavior is harmful and in what contexts it is harmless and even necessary.
4. If we determine the influence on the final decision in certain contexts is cause for concern...finding methods to effectively mitigate or lower the influence of these high-level representations of protected attributes on the final decision while maintaining high accuracy and high utility on the desired task.

The work in this paper tackles the first step–measuring the extent to which various popular pre-trained models are implicitly learning high-level representations of protected attributes. It is important that our readers appreciate the importance of this step and why it is as substantive as the other three steps that fall out from it. As shown in section 3, past research indicates that under some circumstances, pre-trained CNNs are indeed learning high-level concepts even though the models were not explicitly trained to learn these things during the original training task. However, we still do not understand the full landscape of this observed phenomenon. Are we overstating what the pre-trained neural network has learned? Has the model truly learned high-level representations of these concepts or it is simply using low-level representations as proxies for these high-level concepts? Measuring and comparing different models' perceived ability to implicitly learn high-level concepts such as protected attributes during the training process will give us a better understanding of the behavior we are seeing, and lead towards insights that are not tightly coupled to a particular dataset or task. Jumping to steps 2-4 without due diligence on step 1 is tantamount to studying the effect before understanding the cause. The formalization in step 1 is necessary to prevent potential dead-ends, missteps, and incorrect assumptions in the later steps.

5 Methodology

The goal of this paper is to formally examine what high-level attributes popular pre-trained models are learning implicitly, i.e., attributes other than the ones they are trained for, especially protected attributes. Our aim is to formally quantify the extent to which popular, pre-trained CNNs used for visual recognition tasks are learning proxies/protected attributes that they were not explicitly trained on. In this section we discuss the experimental setup used to determine to what extent VGG-16, ResNet-50, and Inception v3, models pre-trained on ImageNet, picked up on three high-level attributes of interest: age, gender, and race. We also used this same experimental setup to conduct a comparative analysis between models pre-trained on ImageNet and those pre-trained on VGGFace and VGGFace2 (VGGFace/2). Models pre-trained on VGGFace use the VGG-16

CNN architecture and models pre-trained on VGGFace2 use the ResNet-50 architecture, as described in the original papers [42, 8]. By comparing the results achieved by models pre-trained on ImageNet against those trained on VGGFace/2 we seek to answer the following question:

- Does learning the visual structure of the world (ImageNet) as opposed to learning the visual structure of faces (VGGFace/2) encode more or less information about age, gender, and race? And if so, to what extent?

5.1 MORPH Dataset

We used the academic, non-commercial version of the MORPH dataset for this project. The MORPH dataset contains a total of 55,134 unique images of 13,618 different people ages 16 to 77 [47]. The image set, composed of mugshots taken from 2003 to 2007, contains images of 11,459 men and 2,159 women. There are 42,589 facial images of African-Americans, 10,559 of European-Americans, 154 Asian, 1,769 Hispanic, and 63 images categorized racially/ethnically as "Other." Each image is labeled with a subject identifier, age of the person on the day the image was taken, the subject's race/ethnicity, and the subject's gender. We used this dataset because of the large age range represented, the uniformity of the images and uncluttered backgrounds, and because of the high representation of African-Americans, a minority group that generally enjoys a lack representation in other more popular datasets.

5.2 Data Preprocessing

We preprocessed the data by resizing the images to $224 \times 244 \times 3$ for VGG-16 and ResNet-50 and to $299 \times 299 \times 3$ for Inception v3. Using the eye-coordinates metadata that comes bundled with the MORPH image set, we used the The PhD Face Recognition Toolbox [56] to horizontally align the eyes, crop out hair, and create geometrically normalized face areas (see Figure 1). We then created three text files containing three disjoint sets of subject ids: a train set, validation set, and test set of sizes 8,170, 2,723, and 2,724 respectively. We made the sets disjoint so that the test and validation sets would not contain images of subjects who already appeared in the training set. The task of predicting a person's age is a regression task. To convert the problem into a classification task, we discretized the age output labels to five bins—bin 0 (ages $<$20), bin 1 (ages 20-35), bin 2 (ages 35-50), bin 3 (ages 50-65), and bin 4 (ages 65+). We relabeled the age ground truth labels with the appropriate bin indices.

Figure 1: Our pictures before and after normalization and cropping.

5.3 Models

For this experiment we used the VGG-16, ResNet-50, and Inception v3 CNN architectures pre-trained on ImageNet, and used the VGG-16 architecture pre-trained on VGGFace and the ResNet-50 architecture pre-trained on VGGFace2. We picked these models because they are some of the most popular models used for transfer learning and are supported by various neural network and machine learning libraries. VGG-16 has 16 layers and performs 3×3 convolutions and 2×2 pooling throughout the entire model [53][38]. ResNet-50 is composed of special skip connections and makes heavy use of batch normalization [24, 26]. Inception v3 was able to reduce the number of parameters in the network by developing an Inception Module [58, 38].

Models pre-trained on ImageNet were trained for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [51] classification task. There are 1000 object classes and approximately 1.2 million training images. VGG-16 models pre-trained on the VGGFace dataset were trained for facial recognition. The dataset contains 2.6 million images of 2,622 celebrities **??**. ResNet-50 architectures pre-trained on the VGGFace2 dataset were also trained for facial recognition tasks. The VGGFace2 dataset contains 3.31 million images of 9,131 celebrities, and is supposed to exhibit larger variations in pose, age, illumination, and ethnicity [8].

5.4 Experimental Set-Up

For each model type, we trained three attribute classifiers—age, gender, and race. We trained the attribute classifiers on six subsets of the training set. Recall that we created a training set of 8,170 subject ids, which contains a total of 33,896 corresponding subject facial images. We created five additional smaller subsets of

the training set, containing 200, 500, 1,000, 2,000, and 4,000 subject ids. These smaller training sets contain 759, 1,937, 3,965, 8,262, and 16,604 images, respectively. Each set of smaller subject ids is a random subset of each larger ids; for example, the subset of 200 ids is a random subset of the 500 ids, which is a random subset of the 1000 ids, and so on. We ensured that we used the exact same train, validation, and test sets for each model type to ensure a fair comparative assessment of classification accuracies across models.

The main motivation behind training on small training sets is that we were particularly interested in how these classifiers perform in very low data settings. For those models trained on ImageNet, if the classifiers generated using transfer learning achieve high accuracy in this very low data setting, then it might be an indication that the original pre-trained model implicitly learned the particular high-level attribute during training for ILSVRC [51]—a classification task that had nothing to do with classifying a person's age, race, or gender. We can then compare these results with those achieved by models pre-trained on VGGFace/2, models that were trained specifically on millions of facial images for the purpose of facial recognition. Intuitively, we expect models pre-trained on VGGFace/2 to perform better on the attribute classification tasks, but how close do models pre-trained on ImageNet get to matching their performance?

We used all three transfer learning methods to train the models: feature extraction, fine-tuning with frozen lower layers, and end-to-end fine-tuning. We are particularly interested in 1) how good the classification accuracy is when using a scheme that manipulates the original pre-trained model as little as possible, i.e., the feature extraction technique and 2) understanding to what extent manipulating the model more and more leads in higher classification accuracies. We also trained the attribute classifiers from scratch as a baseline.

Realizing that the MORPH dataset is heavily imbalanced for gender and race, and fearing that our learned classifiers may simply be predicting male each time, for example, we ran the entire experiment a second time using a resampling strategy during training.

5.5 Training Procedure

We used the same training configurations for all models. Specifically, we trained for 8 epochs using minibatch SGD with a momentum of 0.9, with minibatch size of 128 and an initial learning rate of 0.001. The learning rate decayed on a fixed schedule, specifically after 5 and 7 epochs. Each minibatch was sampled uniformly from the training set and was rescaled appropriately to fit the input of the CNN (i.e. 224×224 for VGG-16). For data augmentation, we apply random horizontally flipping to each image in the minibatch with a probability

of 0.5. We subtracted the mean of the ImageNet data and VGGFace/2 datasets from our training set depending on which models we were training. Each model was trained using one NVIDA P-100 GPU. The overall training process took approximately 1-2 hours for the entire training set, depending on the model used.

6 Results & Discussion

The results of our experiments are shown in Tables 1-3 (models trained on ImageNet) and Tables 7-9 (models trained on VGGFace/2). Overall, certain trends exist for all three of the VGG-16, ResNet-50, and the Inception v3 models. First, we find that training a CNN from scratch usually produces the worst results. This property aligns with previous work [67], which finds that finetuning a pretrained model on a large-scale dataset often results in improved performance. However, we notice that training more parameters of these networks often results in reduced performance when the number of training samples is low, compared to an SVM trained on extracted features. We attribute this property to two reasons: (1) CNNs with more learnable parameters often have a tendency to overfit on small numbers of training samples and (2) the features of pretrained CNNs encode information about protected attributes. The strength of this information is alarmingly high, as strong predictors can be trained from a small number of training examples and also not much lower than one achieved by a finetuned model. Results of from the second iteration of our experiments using data resampling during training are shown in Tables 4-6 for ImageNet models and 10-12 for VGGFace/2.

To visualize how protected attributes are encoded in the pretrained models, we run t-SNE [39] on the extracted features of each pretrained model. As Figures 2 and 3 show, the protected attributes of race, gender, and age are highly segregated, providing further evidence that pretrained models naturally encode this information. In particular we notice that the attributes using features extracted from models pre-trained on VGGFace/2 are better segregated. Curiously, we notice two distinct subgroups emerge for gender and age using models pre-trained on VGGFace (VGG-16 architecture) and VGGFace2 (ResNet-50 architecture), respectively.

We visualize the performance of ResNet-50 architectures pre-trained on ImageNet against the performance achieved by ResNet-50 architectures pre-trained on VGGFace2 in Table 4. The same comparison but using data resampling during training is shown in Table 5. A comparison between VGG-16 architectures pre-trained on ImageNet and those pre-trained on VGGFAce is shown in Table 6 and the resampled data iteration is shown in Table 7. We observe that starting at 500 training IDs, the performance of models pre-trained on ImageNet typically catches up with the performance of those pre-trained on VGGFace/2, at which point the

difference in accuracies between the two models is relatively small. Looking at the age classifier comparision graphs, it appears that learning the visual structure of the world helps classify age better than VGGFace2. This is interesting because one of the goals of VGGFace2 was to increase the age diversity of their dataset [8]. Looking at the race SVM classifiers trained on features extracted from models pre-trained on VGGFace/2, we observe that the classification accuracy barely increases as we increase the number of training samples. This suggests that features extracted from models pre-trained on VGGFace/2 seem to have encoded some notion of race. Whether the CNN has actually learned a high-level representation of race or is merely using low-level representations as proxies for race is left for immediate follow-up work. We also observe that while SVM classifiers for race using features extracted from models pre-trained on VGGFace/2 outperformed those using features extracted from models pre-trained on ImageNet, the opposite is true when using fine-tuning transfer learning techniques. This raises the question of whether ImageNet features are most robust to fine-tuning for race classification.

Recall that one of the primary goals of this paper was to develop a dataset agnostic metric for quantifying how well a pre-trained model might have implicitly learned representations corresponding to high-level concepts, in this case, the protected attributes age, gender, and race. The metric we use is the number of training samples required to achieve the same accuracy across models on a particular task. For example, based on our results, a ResNet-50 CNN pre-trained on ImageNet required 5 times the number of subject IDs as a ResNet-50 CNN pre-trained on VGGFace2 during training to achieve an accuracy of $\sim$95% using end-to-end finetuning. The full table of results can be found in Tables 13-15.

While we framed prediction of the age attribute as a classification problem for training the neural networks, predicting age is more naturally formulated as a regression task rather than a classification task. Thus, using the features extracted from the models pretrained on ImageNet and VGGFace/2, we used the features to train regression models for real valued age prediction. We used the SciKit Learn Python library [43] to learn both Elastic Net [73] and Lasso [59] models. We chose the Lasso model because it uses ℓ_1 prior as regularizer, which offers automatic dimensionality reduction, and chose the Elastic Net model because it uses a combined approach, using ℓ_1 and ℓ_2 priors as regularizer. That said, Elastic Net performed better on the age prediction task, so we present those results here. In Table 8 we show prediction error plots for VGG-16 architectures pre-trained on the ImageNet and VGGFace datasets. In Table 9 we show prediction error plots for ResNet-50 CNNs pre-trained on ImageNet and VGGFace2. These graphs plot the predicted values against the observed

Model Type	Training Size	SVM	Finetune (Freeze)	Finetune (E2E)	Scratch (E2E)
VGG-16	200	91.8	81.7	84.0	76.3
	500	92.9	92.5	94.2	84.3
	1000	93.3	94.1	95.5	89.8
	2000	93.6	95.1	96.2	93.2
	4000	94.0	95.9	96.8	95.2
	8170	94.6	96.5	97.3	96.5
ResNet-50	200	93.7	87.6	88.8	83.0
	500	95.1	93.5	94.6	85.5
	1000	95.4	95.0	95.5	90.6
	2000	96.1	96.0	96.3	93.8
	4000	96.5	96.3	96.8	94.3
	8170	96.9	96.9	97.4	95.4
Inception v3	200	93.3	83.6	84.9	83.9
	500	94.3	92.9	93.9	87.6
	1000	95.0	94.8	95.2	91.0
	2000	95.1	95.8	96.0	94.2
	4000	95.3	96.5	96.7	94.9
	8170	96.2	97.2	97.5	95.6

Table 1: A table which shows the accuracy for different race classifiers using models pre-trained on ImageNet. SVM refers to the accuracy achieved when an SVM was trained on features extracted from an SVM model. Finetune (Freeze) refers to the accuracy achieved after finetuning a ImageNet pretrained CNN, and freezing the weights of layers up to the penultimate convolutional block. Finetune (E2E) refers to the accuracy achieved after finetuning a pretrained CNN end to end. Finally, Scratch (E2E) refers to the accuracy achieved after training the CNN from scratch end-to-end. Each number after the model type refers to the number of individuals (or subjects) used in the training set, with each individual having on average 4 different images.

values. It is clear from the prediction error plots that features extracted from models pre-trained on VGGFace/2 are better at predicting age than those extracted from models pre-trained on ImageNet. This is based on the stronger linear correlation observed between the model's predictions and its actual results, as compared to that of models pre-trained on ImageNet. Mean squared error (MSE) scores and R^2 scores for each of the models shown can be found in the table captions. Models trained using features extracted from CNN's pre-trained on VGGFace/2 consistently achieved lower MSE scores and higher R^2 scores over models trained using features extracted from CNN's pre-trained on ImageNet. This is interesting given that using the same features to train an SVM for age classification resulted in models pre-trained on ImageNet doing better than those pre-trained on VGGFace. Future work is underway to understand why this is the case.

7 Future Work

To confirm our results, we plan to experiment with other datasets besides MORPH, such as Labeled Faces in the Wild (LFW) and MS-Celeb-1M [25, 22]. Although we have demonstrated the inherent nature of pre-trained models to encode information about protected attributes without specific training, we should develop methods to guard against discriminatory practices. For example, adversarial losses [21] can be leveraged to

Model Type	Training Size	SVM	Finetune (Freeze)	Finetune (E2E)	Scratch (E2E)
VGG-16	200	89.2	83.9	83.9	83.9
	500	90.4	86.1	83.9	83.9
	1000	91.3	92.3	95.9	84.8
	2000	91.7	94.5	97.2	89.3
	4000	92.6	96.0	98.3	93.9
	8170	93.9	97.0	98.7	96.7
ResNet-50	200	93.0	83.9	83.9	83.9
	500	95.1	89.2	91.9	83.9
	1000	95.7	92.7	95.1	83.9
	2000	96.6	95.2	97.0	84.0
	4000	97.1	96.4	98.0	87.0
	8170	97.1	97.3	98.4	92.7
Inception v3	200	92.3	83.9	83.9	83.9
	500	93.8	87.0	88.9	84.0
	1000	94.3	93.6	94.6	84.3
	2000	94.9	95.9	96.4	85.7
	4000	95.7	97.4	97.9	89.9
	8170	96.3	98.2	98.6	94.6

Table 2: A table which shows the accuracy for different gender classifiers using models pre-trained on ImageNet. The columns are the same as in Table 1.

Model Type	Training Size	SVM	Finetune (Freeze)	Finetune (E2E)	Scratch (E2E)
VGG-16	200	53.2	53.2	46.5	42.3
	500	51.5	55.6	57.5	42.3
	1000	52.1	58.8	63.0	44.0
	2000	54.8	60.0	67.0	54.2
	4000	57.7	61.9	69.6	60.6
	8170	60.3	63.6	71.6	69.0
ResNet-50	200	54.0	56.2	56.4	38.9
	500	58.7	58.9	59.8	42.3
	1000	60.5	61.9	64.7	47.5
	2000	62.7	66.5	68.2	48.5
	4000	64.6	68.3	70.7	54.1
	8170	66.5	70.5	72.3	57.5
Inception v3	200	53.6	54.4	55.1	43.1
	500	54.5	58.3	59.3	43.0
	1000	57.0	62.3	63.0	45.4
	2000	60.8	67.3	68.8	49.8
	4000	63.1	69.9	70.7	54.8
	8170	66.0	70.7	72.3	63.3

Table 3: A table which shows the accuracy for different age classifiers using models pre-trained on ImageNet. The columns are the same as in Table 1.

Model Type	Training Size	SVM	Finetune (Freeze)	Finetune (E2E)	Scratch (E2E)
VGG-16	200	91.8	82.7	84.6	75.6
	500	92.9	87.1	87.0	74.2
	1000	93.3	88.8	93.1	84.4
	2000	93.7	92.5	95.2	91.5
	4000	94.0	94.5	96.4	94.0
	8170	94.9	95.7	97.1	95.8
ResNet-50	200	93.7	88.9	92.2	42.4
	500	95.1	90.7	94.2	75.6
	1000	95.3	93.5	95.3	81.0
	2000	96.0	95.1	96.3	86.4
	4000	96.3	96.0	96.8	91.5
	8170	96.8	96.5	97.2	94.2
Inception v3	200	93.3	88.0	88.7	81.2
	500	94.3	91.4	91.4	76.5
	1000	95.0	93.6	93.8	81.0
	2000	95.2	95.5	96.0	86.4
	4000	95.2	96.4	96.7	91.5
	8170	95.8	96.7	97.1	94.2

Table 4: A table which shows the accuracy for different race classifiers using models pre-trained on ImageNet. The data were resampled during training to account for the heavily imbalanced dataset. The columns are the same as in Table 1.

Model Type	Training Size	SVM	Finetune (Freeze)	Finetune (E2E)	Scratch (E2E)
VGG-16	200	89.2	86.2	84.5	16.0
	500	90.5	88.9	93.2	67.0
	1000	91.2	91.5	96.3	84.4
	2000	91.9	93.6	97.9	91.5
	4000	92.1	95.4	98.3	95.4
	8170	92.5	96.8	98.6	97.2
ResNet-50	200	93.0	84.2	88.0	24.3
	500	95.3	90.7	93.3	43.1
	1000	95.8	92.6	95.7	66.0
	2000	96.6	95.0	96.9	78.6
	4000	96.7	96.5	98.0	88.0
	8170	97.1	97.3	98.5	93.8
Inception v3	200	92.3	85.3	86.6	66.0
	500	93.8	89.4	91.0	67.6
	1000	94.6	94.2	95.4	72.6
	2000	95.1	95.9	96.9	82.0
	4000	95.8	97.4	98.1	90.1
	8170	95.6	98.1	98.6	94.1

Table 5: A table which shows the accuracy for different gender classifiers using models pre-trained on ImageNet. The data were resampled during training to account for the heavily imbalanced dataset. The columns are the same as in Table 1.

Model Type	Training Size	SVM	Finetune (Freeze)	Finetune (E2E)	Scratch (E2E)
VGG-16	200	53.2	40.4	27.3	0.2
	500	51.7	36.4	31.5	34.6
	1000	51.8	40.9	50.6	31.3
	2000	52.9	47.7	54.3	45.5
	4000	56.5	51.0	62.0	55.2
	8170	55.3	52.7	65.5	62.0
ResNet-50	200	54.0	43.6	50.8	10.6
	500	58.1	53.6	54.5	29.6
	1000	59.8	58.2	63.8	24.3
	2000	60.1	63.6	68.1	34.5
	4000	62.2	66.3	69.9	50.0
	8170	63.7	68.9	71.4	57.0
Inception v3	200	53.6	37.9	46.7	30.6
	500	54.5	49.7	47.5	31.7
	1000	56.3	58.0	60.7	32.6
	2000	58.3	65.0	66.7	40.7
	4000	60.7	69.0	70.0	46.4
	8170	62.3	71.2	71.7	55.9

Table 6: A table which shows the accuracy for different age classifiers using models pre-trained on ImageNet. The data were resampled during training to account for the heavily imbalanced dataset. The columns are the same as in Table 1.

Model Type	Training Size	SVM	Finetune (Freeze)	Finetune (E2E)	Scratch (E2E)
VGG-16	200	95.8			76.3
	500	96.1			84.3
	1000	96.3			89.8
	2000	96.2			93.2
	4000	96.4			95.2
	8170	96.4			96.5
ResNet-50	200	96.7	93.6	95.4	83.0
	500	97.2	95.2	96.4	85.5
	1000	97.1	95.2	97.0	90.6
	2000	97.4	95.5	97.5	93.8
	4000	97.3	95.2	97.3	94.3
	8170	97.9	95.1	97.8	95.4

Table 7: A table which shows the accuracy for different race classifiers using models pre-trained on VGGFace (VGG-16 architecture) and VGGFace2 (ResNet-50 architecture). The columns are the same as in Table 1. Training using freezing and E2E finetuning did not converge, so we do not report those accuracies.

Model Type	Training Size	SVM	Finetune (Freeze)	Finetune (E2E)	Scratch (E2E)
VGG-16	200	94.2			83.9
	500	94.8			83.9
	1000	95.4			84.8
	2000	96.2			89.3
	4000	96.4			93.9
	8170	96.6			96.7
ResNet-50	200	95.5	94.1	92.9	83.9
	500	97.1	97.2	96.6	83.9
	1000	97.7	97.0	97.5	83.9
	2000	98.2	97.2	98.1	84.0
	4000	98.3	96.7	98.8	87.0
	8170	98.2	96.2	99.1	92.7

Table 8: A table which shows the accuracy for different gender classifiers using models pre-trained on VGGFace (VGG-16 architecture) and VGGFace2 (ResNet-50 architecture). The columns are the same as in Table 1. Training using freezing and E2E finetuning did not converge, so we do not report those accuracies.

Model Type	Training Size	SVM	Finetune (Freeze)	Finetune (E2E)	Scratch (E2E)
VGG-16	200	52.4			42.3
	500	55.6			42.3
	1000	54.6			44.0
	2000	53.2			54.2
	4000	56.6			60.6
	8170	59.7			69.0
ResNet-50	200	54.8	55.6	56.2	38.9
	500	57.0	57.4	42.3	42.3
	1000	58.5	63.4	42.3	47.5
	2000	58.5	64.3	59.9	48.5
	4000	60.4	66.6	69.8	54.1
	8170	60.4	66.3	73.3	57.5

Table 9: A table which shows the accuracy for different age classifiers using models pre-trained on VGGFace (VGG-16 architecture) and VGGFace2 (ResNet-50 architecture). The columns are the same as in Table 1. Training using freezing and E2E finetuning did not converge, so we do not report those accuracies.

Model Type	Training Size	SVM	Finetune (Freeze)	Finetune (E2E)	Scratch (E2E)
VGG-16	200	95.8			75.6
	500	96.1			74.2
	1000	96.3			84.4
	2000	96.2			91.5
	4000	96.4			94.0
	8170	96.4			95.8
ResNet-50	200	96.7	93.3	89.9	42.4
	500	97.2	79.4	93.6	75.6
	1000	97.1	90.4	93.8	81.0
	2000	97.4	85.9	94.3	86.4
	4000	97.3	88.5	95.7	91.5
	8170	97.4	94.8	95.5	94.2

Table 10: A table which shows the accuracy for different race classifiers using models pre-trained on VGGFace (VGG-16 architecture) and VGGFace2 (ResNet-50 architecture). The data were resampled during training to account for the heavily imbalanced dataset. The columns are the same as in Table 1. Training using freezing and E2E finetuning did not converge, so we do not report those accuracies.

Model Type	Training Size	SVM	Finetune (Freeze)	Finetune (E2E)	Scratch (E2E)
VGG-16	200	94.2			16.0
	500	94.8			67.0
	1000	95.4			84.4
	2000	96.2			91.5
	4000	96.4			95.4
	8170	96.6			97.2
ResNet-50	200	95.5	93.7	91.5	24.3
	500	97.1	93.0	93.0	43.1
	1000	97.7	96.1	97.3	66.0
	2000	98.2	97.3	97.8	78.6
	4000	98.3	96.9	98.3	88.0
	8170	98.3	91.0	98.3	93.8

Table 11: A table which shows the accuracy for different gender classifiers using models pre-trained on VGGFace (VGG-16 architecture) and VGGFace2 (ResNet-50 architecture). The data were resampled during training to account for the heavily imbalanced dataset. The columns are the same as in Table 1. Training using freezing and E2E finetuning did not converge, so we do not report those accuracies.

Model Type	Training Size	SVM	Finetune (Freeze)	Finetune (E2E)	Scratch (E2E)
VGG-16	200	52.4			0.2
	500	55.6			34.6
	1000	54.6			31.3
	2000	53.2			45.5
	4000	55.5			55.2
	8170	57.2			62.0
ResNet-50	200	54.8	47.0	49.7	10.6
	500	56.9	45.2	40.7	29.6
	1000	58.3	55.5	50.8	24.3
	2000	57.5	57.3	59.4	34.5
	4000	61.3	57.7	63.8	50.0
	8170	64.7	42.9	63.4	57.0

Table 12: A table which shows the accuracy for different age classifiers using models pre-trained on VGGFace (VGG-16 architecture) and VGGFace2 (ResNet-50 architecture). The data were resampled during training to account for the heavily imbalanced dataset. The columns are the same as in Table 1. Training using freezing and E2E finetuning did not converge, so we do not report those accuracies.

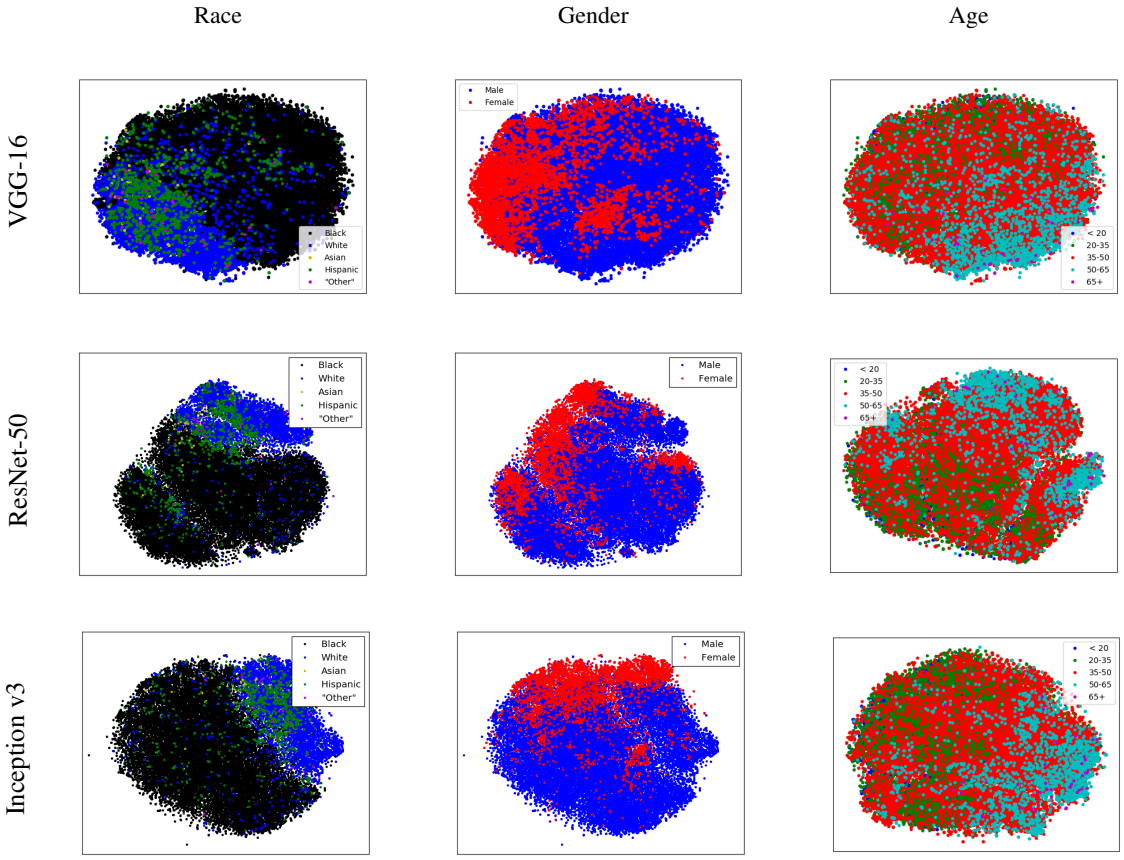

Figure 2: The results of t-SNE on extracted features from each of the ImageNet pretrained VGG-16, ResNet-50, and Inception v3 models. Each t-SNE plot is then colored according to the protected attribute labels of race, gender, and age, showing a clear segregation of the attributes in feature space.

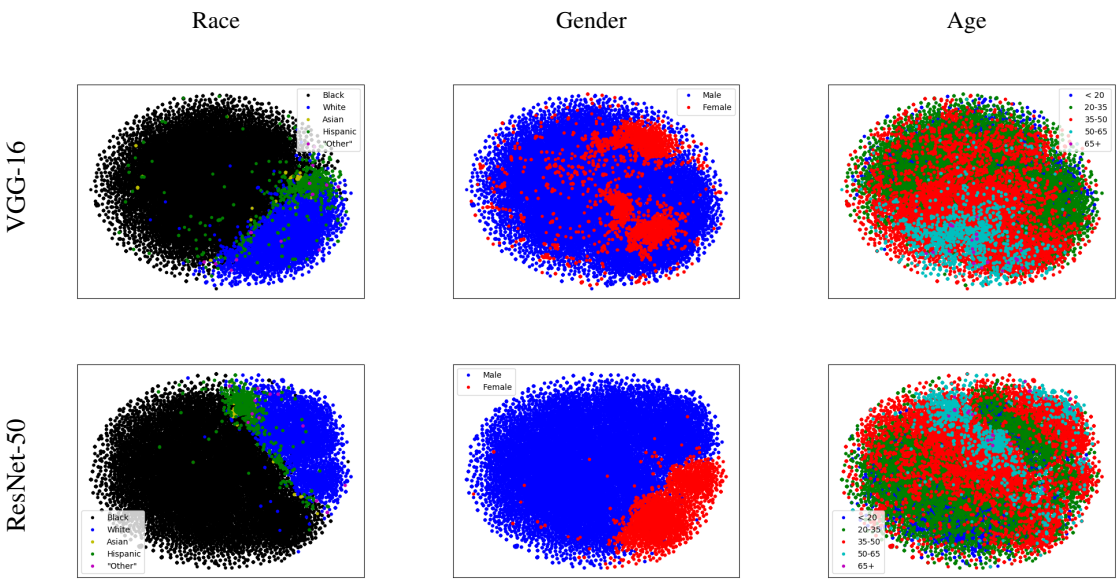

Figure 3: The results of t-SNE on extracted features from each of the VGGFace/2 pre-trained VGG-16 and ResNet-50 models. VGG-16 models were pre-trained on the VGGFace dataset and the ResNet-50 models were pre-trained on the VGGFace2 dataset. Each t-SNE plot is then colored according to the protected attribute labels of race, gender, and age, showing a clear segregation of the attributes in feature space.

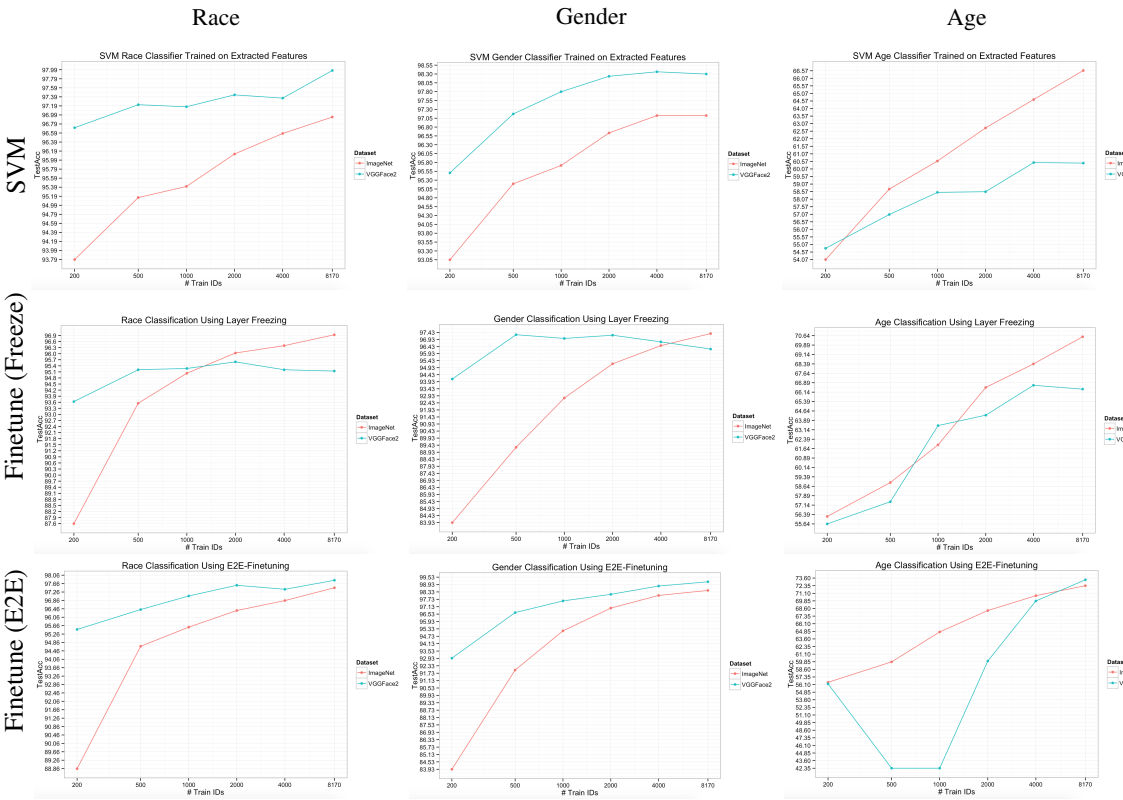

Figure 4: A comparison in attribute classifier performance between ResNet-50 architectures pre-trained on ImageNet (shown in red) and ResNet-50 architectures pre-trained on VGGFace2 (shown in blue) for the various ID sets and transfer learning strategies.

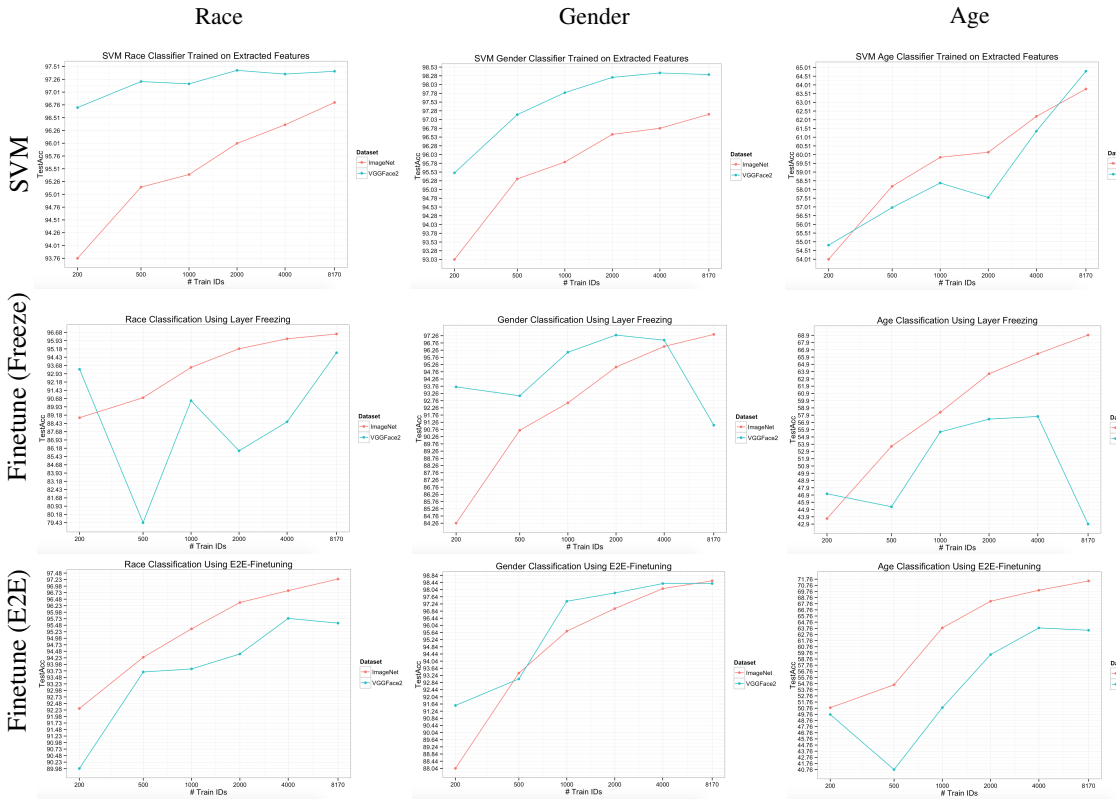

Figure 5: A comparison in attribute classifier performance between ResNet-50 architectures pre-trained on ImageNet (shown in red) and ResNet-50 architectures pre-trained on VGGFace2 (shown in blue) for the various ID sets and transfer learning strategies. The data were resampled during training to account for the heavily imbalanced dataset.

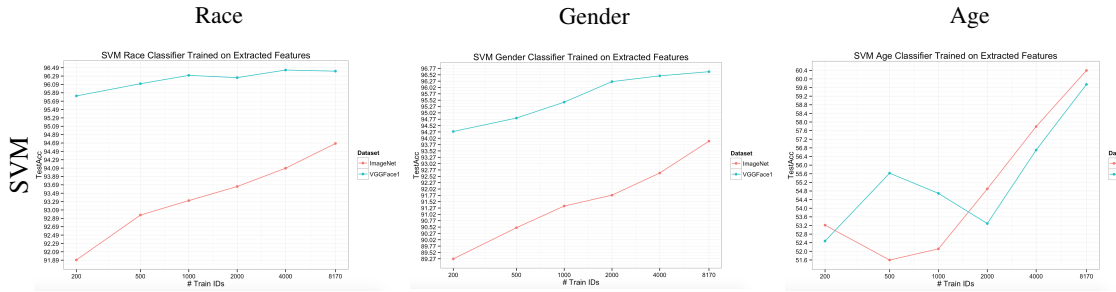

Figure 6: A comparison in attribute classifier performance between VGG-16 architectures pre-trained on ImageNet (shown in red) and VGG-16 architectures pre-trained on VGGFace (shown in blue) for the various ID sets and the feature extraction transfer learning strategy.

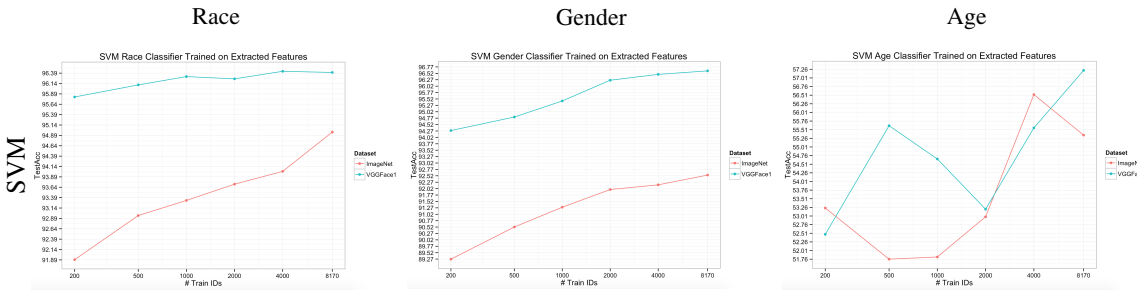

Figure 7: A comparison in attribute classifier performance between VGG-16 architectures pre-trained on ImageNet (shown in red) and VGG-16 architectures pre-trained on VGGFace (shown in blue) for the various ID sets and the feature extraction transfer learning strategy. The data were resampled during training to account for the heavily imbalanced dataset.

Model Type	Balanced	Learning Technique	Accuracy	# IDs (ImageNet)	# IDs (VGGFace/2)	Factor
VGG-16	No	SVM	94.6 vs. 95.8	8170	200	>41
	Yes	SVM	94.9 vs. 95.8	8170	200	>41
ResNet-50	No	SVM	∼96.6	4000	200	20
	No	Freeze	∼93.6	500	200	2.5
	No	E2E	∼95.0	1000	200	5
	Yes	SVM	∼96.7	8160	200	41
	Yes	Freeze	∼90.0	500	1000	2
	Yes	E2E	∼94.3	500	2000	4

Table 13: A table which shows the number of samples required to achieve parity of accuracies between different race classifiers using models pre-trained on ImageNet and models pre-trained on VGGFace/2. The 'Learning Technique' refers to the transfer learning method used to train that particular model, and 'Balanced' refers to whether the data were resampled during the training of the model or not. Neither of the VGG-16 models shown in this table, pre-trained on ImageNet were able to achieve accuracy parity with the VGG-16 models pre-trained on VGGFace. While the model pre-trained on VGGFace, which did not use data resampling, achieved an accuracy of 95.8% using a training set of only 200 subject IDs, the model pre-trained on ImageNet was only able to reach an accuracy of 94.6% despite using the full training set of 8170 IDs. Thus, we conjecture that to achieve an accuracy of 95.8%, the model pre-trained on ImageNet would require a factor of more than 41 times the number of training subject IDs needed by the model pre-trained on VGGFace.

Model Type	Balanced	Learning Technique	Accuracy	# IDs (ImageNet)	# IDs (VGGFace/2)	Factor
VGG-16	No	SVM	∼94.0	8170	200	41
	Yes	SVM	92.5 vs. 94.2	8170	200	>41
ResNet-50	No	SVM	∼97.0	4000	500	8
	No	Freeze	∼97.0	8170	500	16
	No	E2E	∼98.0	4000	2000	2
	Yes	SVM	∼95.4	500	200	2.5
	Yes	Freeze	∼93.0	1000	500	2
	Yes	E2E	∼93.0	500	500	1

Table 14: A table which shows the number of samples required to achieve parity of accuracies between different gender classifiers using models pre-trained on ImageNet and models pre-trained on VGGFace/2. The columns are the same as in Table 13. Looking at the VGG-16 models, which used data resampling during training, we see that the model pre-trained on ImageNet did not reach accuracy parity with the model pre-trained on VGGFace. While the model pre-trained on VGGFace achieved an accuracy of 94.2% using a training set of only 200 subject IDs, the model pre-trained on ImageNet was only able to reach an accuracy of 92.5% despite using the full training set of 8170 IDs. Thus, we conjecture that to achieve an accuracy of 94.2%, the model pre-trained on ImageNet would require a factor of more than 41 times the number of training subject IDs needed by the model pre-trained on VGGFace.

Model Type	Balanced	Learning Technique	Accuracy	# IDs (ImageNet)	# IDs (VGGFace/2)	Factor
VGG-16	No	SVM	$\sim$ equal			
	Yes	SVM	$\sim$53.0	2000	2000	1
ResNet-50	No	SVM	$\sim$58.5	500	1000	2
	No	Freeze	$\sim$66.4	2000	4000	2
	No	E2E	$\sim$60.0	500	2000	4
	Yes	SVM	$\sim$58.0	500	1000	2
	Yes	Freeze	$\sim$58.0	1000	4000	4
	Yes	E2E	$\sim$64.0	1000	4000	4

Table 15: A table which shows the number of samples required to achieve parity of accuracies between different gender classifiers using models pre-trained on ImageNet and models pre-trained on VGGFace/2. The columns are the same as in Table 13. The VGG-16 models, which did not use data resampling during training, had about the same accuracy across training ID sets regardless of whether the model was pre-trained on ImageNet or VGGFace.

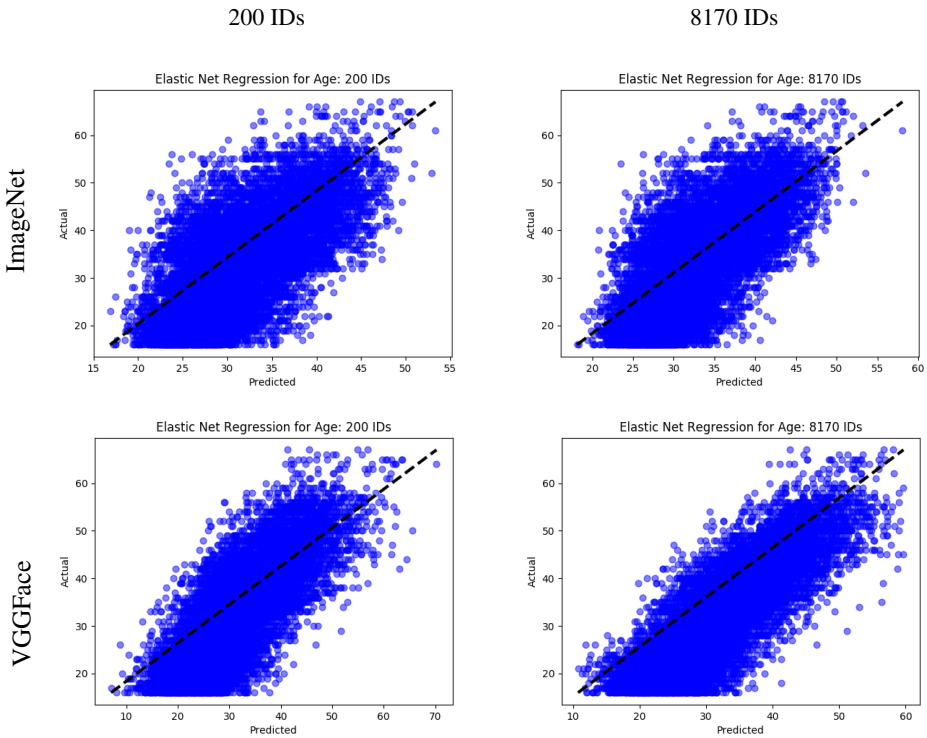

Figure 8: A comparision between prediction error plots for VGG-16 models pre-trained on ImageNet and VGGFace using the smallest set of training IDs and the complete set of training IDs. Features extracted from these models were used to train Elastic Net regression models for real valued age prediction. For the training set of 200 IDs, the model pre-trained on ImageNet got an MSE score of 70.05 and an R^2 score of 0.42. The same model architecture pre-trained on VGGFace achieved an MSE of 49.1 and an R^2 score of 0.59. Using the full set of 8170 IDs, the model pre-trained on ImageNet got an MSE score of 65.32 and an R^2 score of 0.46. The same model architecture pre-trained on VGGFace achieved an MSE of 39.68 and an R^2 score of 0.67.

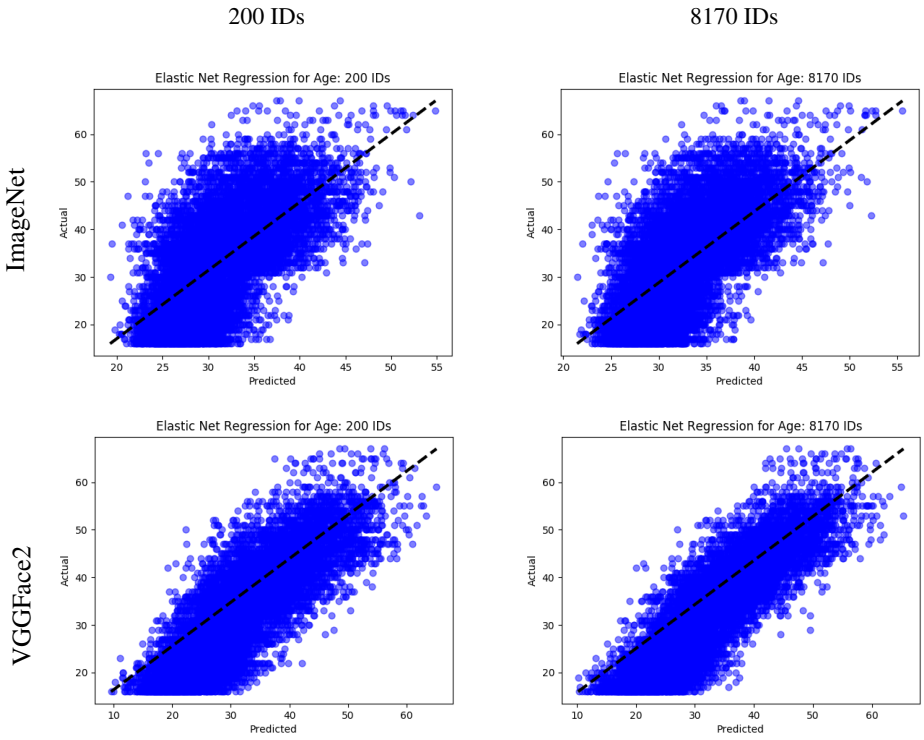

Figure 9: A comparision between prediction error plots for ResNet-50 models pre-trained on ImageNet and VGGFace2 using the smallest set of training IDs and the complete set of training IDs. Features extracted from these models were used to train Elastic Net regression models for real valued age prediction. For the training set of 200 IDs, the model pre-trained on ImageNet got an MSE score of 75.75 and an R^2 score of 0.38. The same model architecture pre-trained on VGGFace2 achieved an MSE of 39.13 and an R^2 score of 0.67. Using the full set of 8170 IDs, the model pre-trained on ImageNet got an MSE score of 73.34 and an R^2 score of 0.39. The same model architecture pre-trained on VGGFace2 achieved an MSE of 30.52 and an R^2 score of 0.75.

learn representations that are not predictive of protected attributes [15]. Moreover, a natural next step is to identify the particular neurons encoding the attributes of interest and analyze how the removal of these neurons affects the pre-trained models' ability to produce accurate results on a task such as facial image recognition. Another variant would be to analyze saliency maps to identify regions of images that strongly contribute a class score [54]. Other future work includes repeating the experiment for the age classification task but using different age bin schemes. We already experimented using different age ranges and saw significant improvement using the age cutoffs presented in this paper over a previous iteration of age range cutoffs where we observed a 15 percent relative error. We saw this large improvement in accuracies across all models and transfer learning schemes. Perhaps the pre-trained models we studied did indeed implicitly learn age well but at a higher granularity. For example, instead of five ages bins we could use three bins to see if we get improved accuracy. It would also be interesting to see if the pre-trained CNNs could classify albinos into their appropriate race categories by learning more granular features besides skin color.

The following list summarizes tasks for future consideration and includes tasks that we are currently working on. Preliminary results are encouraging.

- Develop heuristics to differentiate between learning of high-level representations associated with protected attributes and learning of low-level representations that are simply being used as proxies for these protected attributes.
- When comparing the accuracies achieved by models trained on ImageNet and those trained on VGGFace/2, it is not obvious by looking at the data what these differences are attributed to. Also, it is not immediately clear why models pre-trained on VGGFace/2 are amenable to feature extraction but not always to E2E fine-tuning. Additional experiments are required to understand these results.
- Run the entire suite of experiments multiple times to get error bars on the accuracies.
- Use cross-validation to ensure we have found the most optimal hyperparameters (e.g. number of training epochs, learning rate, dropout probability, decay rate) during training and the transfer learning process.
- In Section 6, we looked at the difference in the number of training images needed to achieve the same accuracy across CNN's pre-trained on ImageNet and those trained on VGGFace/2 for each of the learned attribute classifiers. Run the experiments using other datasets such as LFW and MS-Celeb-1M to see if these computed ratios are dataset agnostic and stable across various datasets.
- Use existing methods in CNN interpretability to shed some light on the high-level concepts being learned in the final layer of the neural network models. This includes current work to identify specific neurons that may be encoding the protected attributes of interest, i.e. age, gender, and race.
- Extend the experiments to learn other protected attributes beyond age, gender and race such as disability status, religious affiliation, pregnancy, and national origin.
- Visualize and provide a semantic explaination for the distinct subgroups that arose in several of the TSNE graphs.
- Run diagnostic and debugging tools for general sanity checking. Some of the accuracies observed may point to instability during the training step.

Finally, the ethical implications of exploiting people caught in the criminal justice system and using their mugshots to build viable computer vision imagesets must also be considered. And overall, the results from our

experiments, which compare models pre-trained on images of celebrities with models pre-trained on images of people on their worst day (mugshots), raises the question of how the high-level conditions under which facial images are taken affects classification of age and gender.

8 Conclusions

In this paper we showed how ResNet-50, VGG-16, and Inception v3, three of the most popular computer vision models pre-trained on ImageNet, exhibit a surprisingly high level of accuracy in determining three different protected attributes, tasks which are seemingly orthogonal to the object classification task they were originally trained for. While the same two CNN architectures, VGG-16 and ResNet-50, pre-trained on VGGFace/2 performed better on the majority of the attribute classification tasks as compared to the same two architectures pre-trained on ImageNet, after 500 subject ids, the performance achieved by models pre-trained on ImageNet was comparable to the performance achieved by the same models pre-trained on VGGFace/2. The evidence we presented in this paper shows that there is a strong possibility that these models implicitly learned how to encode race, gender, and age information during the original pre-training process. While further work needs to be conducted to ascertain whether this is indeed the case as well as understand how this happened, it points to a larger problem on the horizon. Are researchers, hobbyists, and government agencies using and adapting these and similar models for their visual classification tasks without fully understanding how features unbeknownst to them (and especially features that encode protected attributes), are indirectly influencing the outcome of their results (and in potentially biased and harmful ways)?

We do not need to accept non-transparency and bias as the status quo in machine learning and computer vision. In fact, we must not accept it. We have the power and obligation to study these issues scientifically, expose flaws and misuse, and insist on fairness in machine learning and insist on the ethical deployment of computer vision applications for the greater good of society and humanity.

References

[1] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias. *Pro Publica*, 2016.

[2] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition*, 2017.

[3] Y. Bechavod and K. Ligett. Learning fair classifiers: A regularization-inspired approach. *CoRR*, abs/1707.00044, 2017. URL http://arxiv.org/abs/1707.00044.

[4] M. Buda, A. Maki, and M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *arXiv preprint arXiv:1710.05381*, 2017.

[5] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In S. A. Friedler and C. Wilson, editors, *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 77–91, New York, NY, USA, 23–24 Feb 2018. PMLR. URL http://proceedings.mlr.press/v81/buolamwini18a.html.

[6] M. Burgess. Facial recognition tech used by uk police is making a ton of mistakes. http://www.wired.co.uk/article/face-recognition-police-uk-south-wales-met-notting-hill-carnival, May 2018.

[7] F. Calmon, D. Wei, B. Vinzamuri, K. Natesan Ramamurthy, and K. R. Varshney. Optimized pre-processing for discrimination prevention. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3992–4001. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/6988-optimized-pre-processing-for-discrimination-prevention.pdf.

[8] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*, 2018.

[9] A. Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *arXiv preprint arXiv:1703.00056*, 2017.

[10] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq. Algorithmic decision making and the cost of fairness. *arXiv preprint arXiv:1701.08230*, 2017.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[12] S. Denyer. Beijing bets on facial recognition in a big drive for total surveillance. https://www.washingtonpost.com/news/world/wp/2018/01/07/feature/in-china-facial-recognition-is-sharp-end-of-a-drive-for-total-surveillance/?utm_term=.7a5c2a96fadb, 2018.

[13] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 214–226. ACM, 2012.

[14] H. Edwards and A. J. Storkey. Censoring representations with an adversary. *CoRR*, abs/1511.05897, 2015.

[15] H. Edwards and A. J. Storkey. Censoring representations with an adversary. *CoRR*, abs/1511.05897, 2015. URL `http://arxiv.org/abs/1511.05897`.

[16] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

[17] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013.

[18] M. Garcia. Racist in the machine the disturbing implications of algorithmic bias. *World Policy Journal*, 33(4): 111–117, 2016.

[19] C. Garvie, A. Bedoya, and J. Frankle. The perpetual line-up: Unregulated police face recognition in america. `https://www.perpetuallineup.org`, October 2016.

[20] T. Gebru, J. Krause, Y. Wang, D. Chen, J. Deng, and L. Fei-Fei. Visual census: Using cars to study people and society, January 2015. URL `https://www.microsoft.com/en-us/research/publication/visual-census-using-cars-study-people-society/`.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[22] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In *European Conference on Computer Vision*. Springer, 2016.

[23] M. Hardt. How big data is unfair. `https://medium.com/@mrtz/how-big-data-is-unfair-9aa544d739de`, 2014.

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[25] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[26] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[27] H. E. Khiyari* and H. Wechsler. Face verification subject to varying (age, ethnicity, and gender) demographics using deep learning. *Journal of Biometrics and Biostatistics*, 7(4):–, 2016. ISSN 2155-6180. doi: 10.4172/2155-6180.1000323. URL https://www.omicsonline.org/open-access/face-verification-subject-to-varying-age-ethnicity-and-genderdemographics-using-deep-learning-2155-618 php?aid=82636.

[28] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision – ECCV 2012*, pages 158–171, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33718-5.

[29] N. Kilbertus, M. Rojas-Carulla, G. Parascandolo, M. Hardt, D. Janzing, and B. Schölkopf. Avoiding discrimination through causal reasoning. In *Proceedings from the conference "Neural Information Processing Systems 2017.*, pages 656–666. Curran Associates, Inc., Dec. 2017. URL http://papers.nips.cc/paper/6668-avoiding-discrimination-through-causal-reasoning.pdf.

[30] B. F. Klare, M. J. Burge, J. C. Klontz, R. W. V. Bruegge, and A. K. Jain. Face recognition performance: Role of demographic information. *IEEE Transactions on Information Forensics and Security*, 7(6):1789–1801, Dec 2012. ISSN 1556-6013. doi: 10.1109/TIFS.2012.2214212.

[31] J. Kleinberg, H. Lakkaraju, J. Leskovec, J. Ludwig, and S. Mullainathan. Human decisions and machine predictions. Working Paper 23180, National Bureau of Economic Research, February 2017. URL http://www.nber.org/papers/w23180.

[32] P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/koh17a.html.

[33] A. Kovashka, O. Russakovsky, L. Fei-Fei, and K. Grauman. Crowdsourcing in Computer Vision. *ArXiv e-prints*, Nov. 2016.

[34] R. Kramer, A. Young, M. Day, and A. Burton. Robust social categorization emerges from learning the identities of very few faces. *Psychological Review*, 124(2):115–129, 3 2017. ISSN 0033-295X. doi: 10.1037/rev0000048. (c)

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[36] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014.

[37] Y. LeCun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'04, pages 97–104, Washington, DC, USA, 2004. IEEE Computer Society. URL `http://dl.acm.org/citation.cfm?id=1896300.1896315`.

[38] F.-F. Li, A. Karpathy, and J. Johnson. Cs231n: Convolutional neural networks for visual recognition 2016. URL `http://cs231n.stanford.edu/`.

[39] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov): 2579–2605, 2008.

[40] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[41] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks, June 2014. ISSN 1063-6919.

[42] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2015.

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[44] G. Pleiss, M. Raghavan, F. Wu, J. M. Kleinberg, and K. Q. Weinberger. On fairness and calibration. In *NIPS*, pages 5684–5693, 2017.

[45] N. Raval, A. Machanavajjhala, and L. P. Cox. Protecting visual secrets using adversarial nets. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1329–1332. IEEE, 2017.

[46] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, CVPRW '14, pages 512–519, Washington, DC, USA, 2014. IEEE Computer Society. ISBN 978-1-4799-4308-1. doi: 10.1109/CVPRW.2014.131. URL `http://dx.doi.org/10.1109/CVPRW.2014.131`.

[47] K. Ricanek Jr. and T. Tesafaye. Morph: A longitudinal image database of normal adult age-progression. In *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, FGR '06, pages 341–345, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2503-2. doi: 10.1109/FGR.2006.78. URL `https://doi.org/10.1109/FGR.2006.78`.

[48] D. Riccio, G. Tortora, M. D. Marsico, and H. Wechsler. Ega ethnicity, gender and age, a pre-annotated face database. In *2012 IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications (BIOMS) Proceedings*, pages 1–8, Sept 2012. doi: 10.1109/BIOMS.2012.6345776.

[49] H. Rudolph, L. M. Moy, and A. M. Bedoya. Not ready for takeoff: Face scans at airport departure gates. 2017. URL `https://www.airportfacescans.com`.

[50] D. E. Rumelhart, G. E. Hinton, R. J. Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[51] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[52] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, volume 00, pages 618–626, Oct. 2018. doi: 10.1109/ICCV.2017.74. URL `doi.ieeecomputersociety.org/10.1109/ICCV.2017.74`.

[53] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[54] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[55] J. Sokolic, Q. Qiu, M. R. Rodrigues, G. Sapiro, A. Bansal, C. Castillo, R. Ranjan, R. Chellappa, H. A. Le, I. A. Kakadiaris, et al. Learning to identify while failing to discriminate. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[56] V. Struc. The phd face recognition toolbox, 2012. URL https://www.mathworks.com/matlabcentral/fileexchange/35106-the-phd-face-recognition-toolbox. (visited on 2018-01-09).

[57] L. Sweeney. Discrimination in online ad delivery. *Queue*, 11(3):10:10–10:29, Mar. 2013. ISSN 1542-7730. doi: 10.1145/2460276.2460278. URL http://doi.acm.org/10.1145/2460276.2460278.

[58] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[59] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58: 267–288, 1994.

[60] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1521–1528. IEEE, 2011.

[61] U.S. Equal Employment Opportunity Commission. Laws & guidance. URL https://www.eeoc.gov/policy/vii.html. (visited on 2018-01-12).

[62] C. Vondrick, H. Pirsiavash, A. Oliva, and A. Torralba. Learning visual biases from human imagination. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 289–297. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5781-learning-visual-biases-from-human-imagination.pdf.

[63] Y. Wang and M. Kosinski. Deep neural networks are more accurate than humans at detecting sexual orientation from facial images. 2017.

[64] D. Wei, B. Zhou, A. Torralba, and W. T. Freeman. Understanding intra-class knowledge inside CNN. *CoRR*, abs/1507.02379, 2015.

[65] X. Wu and X. Zhang. Automated inference on criminality using face images. *arXiv preprint arXiv:1611.04135*, 2016.

[66] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3320–3328, Cambridge, MA, USA, 2014. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969033.2969197.

[67] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[68] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. In *Deep Learning Workshop, International Conference on Machine Learning (ICML)*, 2015.

[69] M. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision, ECCV 2014 - 13th European Conference, Proceedings*, volume 8689 LNCS of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 818–833. Springer Verlag, 2014. ISBN 9783319105895. doi: 10.1007/978-3-319-10590-1_53.

[70] B. H. Zhang, B. Lemoine, and M. Mitchell. Mitigating unwanted biases with adversarial learning. *CoRR*, abs/1801.07593, 2018.

[71] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv preprint arXiv:1707.09457*, 2017.

[72] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *CoRR*, abs/1412.6856, 2014.

[73] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.