2020

University of North Carolina Wilmington

Master of Science in

Computer Science and Information Systems

Proceedings

https://csbapp.uncw.edu/mscsis

MITIGATING ALGORITHMIC BIAS IN DEEP CONVOLUTIONAL NEURAL NETWORKS

Philip Smith

A Thesis Submitted to the
University of North Carolina Wilmington in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

Department of Computer Science
and
Congdon School of Supply Chain, Business Analytics and Information Systems

University of North Carolina Wilmington

2020

Approved by

Advisory Committee

Dr. Sudip Mittal

Dr. Douglas Kline

Dr. Karl Ricanek
Chair

Accepted by

Dean, Graduate School

TABLE OF CONTENTS

ABSTRACT

Despite the numerous promising advancements in the field of artificial intelligence, algorithmic bias is pervasive in modern deep learning systems, and it is often ignored. Biometric AIs have been identified to contain racial bias, gender bias, age bias, and more. This thesis is a study of the bias that exists in face-based soft biometric systems. Three solutions are proposed for the purpose of mitigating bias. The first solution involves data augmentation as a method of mitigating bias. Trained models are usually biased against dataset minorities, and since data augmentation is essentially a method of manufacturing more training data, it is proposed that data augmentation will help glean more information from these minorities. The second solution involves oversampling parts of the dataset. This gives a model a more even representation across different classes so that error is not simply optimized for the majority. The third solution involves the use of generated data to synthetically help balance the training set. Each of these solutions are explored for their efficacy of bias mitigation. State-of-the-art results are obtained for age, gender, and BMI estimation.

ACKNOWLEDGEMENTS

I started down this line of research in 2018 as part of a NSF REU program during the summer before my final senior semester of my Bachelor's degree. Before then, I had never trained a neural network and only had a vague understanding of how they worked. I had particularly enjoyed a computer science course, *Scientific Computing*, in a prior semester, which drove me to seek out machine learning positions for the summer. I was ecstatic about being accepted into the REU, and it would end up having a much larger impact on my life than I anticipated. Early into the program, we were given the MORPH-II dataset, a collection of mugshots labeled with age, race, gender, and other such features, and told to begin analyzing the dataset while looking for inconsistencies. While we crammed information on traditional statistical classifiers, we were given projects each night to make estimations for the dataset using a set of extracted biologically inspired features. Being a fairly confident programmer, I soon chose to take the deep learning approach to making estimations for age and gender given just a face image. This work led to the publication of my first paper: *Transfer Learning with Deep CNNs for Gender Recognition and Age Estimation.* I had quickly obtained some good results using transfer learning, and that drove me to continue my research to see if I could keep improving the AI predictions. After experimenting with data augmentation and a number of other datasets, I found out that I could improve the performance of an AI while also generalizing it to become more accurate for a wider variety of inputs. It was around this time that I met with Dr. Karl Ricanek, a well-established researcher in the field of biometrics, to discuss topics for a thesis. He led me to realize that, while my techniques improved overall AI performance, the models could actually become more biased against certain subgroups, causing them to underperform for those groups. Since then, I have been working in Karl's I3S lab where I have conducted a number of studies relating to biometric neural networks, model bias analysis, and leveraging available data and labels to improve AI performance. The AIs that we use to make accurate big data

predictions faster than any group of humans can, should not and cannot be systemically biased against certain portions of the population. This thesis is the culmination of my work towards mitigating such algorithmic bias. I would like to thank Dr. Cuixian Chen and Dr. Yishi Wang for enabling me to commence this AI research under NSF grant DMS-1659288. I would like to thank Dr. Karl Ricanek for providing me with many resources and opportunities to further pursue this work. I'd like to thank my fellow lab members for challenging me and inspiring new ideas every day, and I would also like to thank Dr. Gene Tagliarini for sparking my interest in data science and artificial intelligence.

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Bias is something that has long divided humankind. Neighbors have started wars over small cultural differences. Day-to-day interactions between people are modified based on gender and skin color. None of us are immune to it, we can only strive to do the best we can to not allow biases to affect our decisions. Recently, a number of large tech companies such as Amazon, Microsoft, and IBM, have placed a temporary ban on the use of face recognition technology by law enforcement due to the misuse of the software in due process. Face recognition, while an impressively powerful tool, is a technology with known biases [2]. AI has the potential revolutionize nearly every industry, but as we begin to rely more and more upon it, we need to ensure that it is studied in a way that it will do no harm to mankind. Systemic bias is a travesty which has plagued, not only our society, but many other societies as well. Algorithms cannot, and should not, be biased against or favor certain portions of the population. Succinctly:

> Algorithmic bias describes systematic and repeatable errors in a computer system that create unfair outcomes, such as privileging one arbitrary group of users over others. Bias can emerge due to many factors, including but not limited to the design of the algorithm or the unintended or unanticipated use or decisions relating to the way data is coded, collected, selected or used to train the algorithm [3].

Taking this into consideration, it doesn't matter how high the overall accuracy of an algorithm is, if it still fails considerably in certain situations. AI is a very controversial topic today because it is sparsely available, largely misunderstood, and there are many new ethical and philosophical issues to consider. This thesis is a study of algorithmic bias, and proposes solutions for mitigating such bias. Specifically, this work takes a look at face-analytic technologies and the biases ingrained within it. Face-image models for age, gender, and body mass index (BMI) estimation have been the subject of numerous studies in the

past, but in these studies, bias is often ignored. A wide variety of feature extraction methods and statistical classifiers have been used to try to get accurate analytic results [4]. This work, however, focuses specifically on the use of deep convolutional neural networks (DCNNs) to generate an analysis of a face image: specifically for the inference of age, gender, and BMI. In this thesis, I will provide historical perspectives on the body of work for facial analytics with respect to age, gender, and BMI. I will provide an overview of the research literature regarding facial analytic bias in order to provide a definition of bias and establish the overall objective of this body of research.[1] I will review prior work that investigates the mitigation of performance differences through data, which is the focus of this work. This work does not focus on other techniques for bias reduction as a point of focus. This thesis proposes three novel solutions for mitigating bias in neural networks during the training phase. Using large public datasets, these solutions are experimentally explored to observe their effects with regard to age, race, gender, and BMI.

[1] Although face recognition is discussed, this work does not address the specific tasks of face recognition, verification, or clustering.

BACKGROUND

Defining Bias

Historically, bias has had many different definitions in the realm of neural networks. The oldest definition originated with the concept of the Perceptron [5]. In 1957, Frank Rosenblatt of Cornell Aeronautical Lab demonstrated that machines could learn to recognize things that exist in the "phenomenal world" with a small simulation of the brain composed of "perceptrons" rather than neurons. Perceptrons operated according to the weights and bias associated with the perceptron as seen in formula 1 below.

$$f(x) = \begin{cases} 1 & \text{if} \sum_{i=0}^{j-1} w_i x_i + b > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

In formula 1, x is the input to the perceptron while w is the weight associated with each input, and j is the total number of weights and inputs. b is the bias that is simply added to the sum of the weight and input products which sometimes causes the perceptron to activate. Modern day neurons operate in an extremely similar manner, often with the aid of an activation function such as ReLU (rectified linear unit). ReLU is defined as $\max(f(x), 0)$. An example of a neuron can be seen in figure 1 below.

Another traditional definition of bias deals with the error in a network's output. In [6], bias is defined as:

$$\{E_D[f(x; D)] - E[y|x]\}^2 \tag{2}$$

where E_D is the expected output given the training set D and $E[y|x]$ is the error of example x given label y. As stated in [6], "If, on the average, $f(x; D)$ is different from $E[y|x]$, then $f(x; D)$ is said to be biased as an estimator of $E[y|x]$." However, this definition of bias in now obsolete because modern neural networks can almost fully memorize any training data,

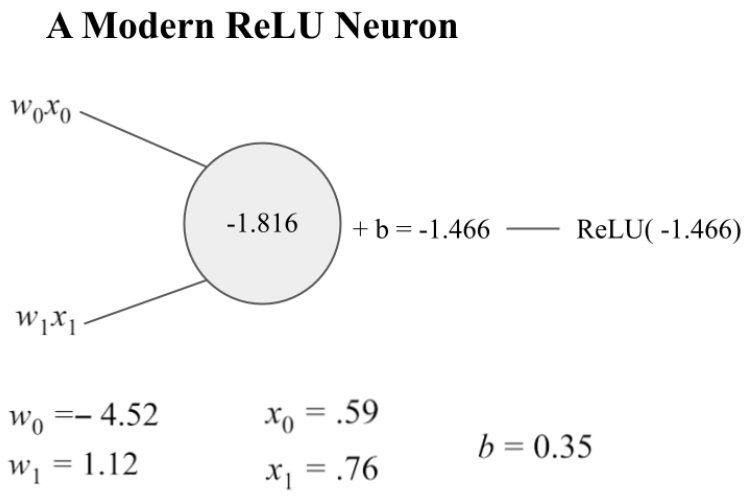

Figure 1: The inputs to the neuron are multiplied by their respective weights and the bias is added. The activation function is applied to produce the output of the neuron which, in this instance, is zero.

so held-out test sets are used to estimate error.

In this thesis, bias is understood to mean that there is some variation in performance between different demographic groups such as race, age, or gender. For example, an age estimator that produces an average error of 3 years overall, but 9 years for people under age 10, can be said to be negatively biased against children 10 or younger.

Dataset Bias

For a long time, one of the main sources of algorithmic bias has been the dataset itself. In 2011, Torralba and Efros of MIT and CMU perform a cross-dataset examination of image classifier performance to reveal the bias that exists within datasets [7]. They studied 12 datasets and found that they are able to train a classifier that can determine which of the datasets an image belongs to with 39% accuracy – 31% higher than random guessing. The authors make the striking point that there appeared to be very little cross-dataset generalization occurring in academic research published during that time period. For example, a classifier trained to recognize cars on one dataset cannot easily recognize cars from a different dataset. The authors identify and detail 4 types of dataset bias. The first, *selection bias*,

is introduced into a dataset because particular types of scenes are preferred such as street or nature scenes. The second, *capture bias*, is introduced by photographers tending to take photos of objects in similar ways. The third, *category bias*, stems from poorly separated labelling schemes such as "grass" vs. "lawn" or "painting" vs. "picture". The fourth, *negative set bias*, is an effect produced by imbalanced datasets. The authors also claim that machine-aggregated data are typically less biased because *human bias* is introduced into the dataset when hand-picking examples. Model bias may also be introduced by the existence of *dirty data* in a dataset. If a dataset contains a small subgroup that contains many "dirty" or mislabeled images, the model's performance for that subgroup might worsen greatly.

Other Sources of Algorithmic Bias

Aside from problems with the dataset, bias can be introduced into an algorithm via other means. One such source of bias that is identified in this work is *classifier bias*. Given a clean and balanced dataset, two classifiers might produce drastically different results for certain classes or demographics. For example, the adaboost classifier might produce different results than logistic regression, and even if they have the same accuracy, they might misclassify different sets of examples. The same can be said about neural networks, which are growing increasingly more complex, but they are still essentially just classifiers. Not much work has been put into identifying neural network architectural components that cause bias or loss of generalization. Another source of bias is *misappropriation bias*. An algorithm could be unbiased and yielding very accurate results, but if an end-user finds a way to misuse the algorithm, then bias could still result from an unbiased algorithm. Timnit Gebru, who is a Google AI researcher, identifies a final type of bias known as *automation bias*. The example she uses to describe automation bias is a scenario in which a law enforcement officer is trying to identify a criminal in a surveillance photo. The officer might believe very strongly that the mugshot matched by a face recognition system is not the same person who is in the surveillance footage, but if the algorithm reports 99% confidence in the match, the officer's

opinion could change, and the wrong person could be arrested.

Bias in Face Recognition

Governments, corporations, and academic institutions have all reported bias issues in face-based image processing technologies. In 2019, NIST conducted a large scale evaluation of 189 one-to-one and one-to-many face recognition algorithms. *Face verification* is one-to-one face matching where two face-images are fed through a network which should report "match" if they are of the same person, or "non-match" if they are different people. *Face recognition* is a one-to-many procedure in which subjects are enrolled in a template with several images and then the subject is identified using an unseen face-image. NIST conducted this study using 18.27 million images of 8.49 million subjects with a particular emphasis on the demographic performance of the algorithms. The algorithms are evaluated for bias by examining the false positive and false negative match rates by race, gender, and age. The study found that most algorithms suffer from high false match rates (FMR) in countries like Somalia and Kenya. Additionally, countries with similar demographics would have similarly poor cross-culture FMRs, such as Vietnam and Thailand. This means that a Vietnamese person might have a high chance of being confused with a Thai person and vice versa. Some algorithms were clearly biased by the available training data. Chinese algorithms generally performed better on Asians than other ethnicities. The Yitu-003 algorithm was one of the most accurate, and also had the most uniform FMR. For African women, however, the Yitu-003 FMR was as high as 0.3%, failing on roughly 1 in 300 individuals. It was additionally demonstrated that face recognition performed worse on women in nearly every culture, and the FMR for children and elderly people was high when paired with an imposter from the same age group. *Imposters* are simply images of a different subject meant to try to trick the network into a false match. This thesis does not address bias in face recognition, but it is currently a very controversial topic, and face recognition bias has led to the investigation of bias in various other machine learning models, products, and algorithms.

Age and Gender Estimation

Age and gender estimation from face-images are widely studied tasks due to the growing need for identifying individuals and gleaning soft biometrics. *Soft biometrics* are estimated rather than physically monitored or calculated. For example, age can be inferred from an image of an individual, but the inferred age might differ from the individual's actual chronological age. One of the first datasets assembled to study computational age and gender estimation was FG-NET (face and gesture net). It was released in 2004 and contains 1002 images of 82 individuals at different points in their lives. FG-NET was most commonly used as a benchmark for evaluating and comparing age and gender models. Age results are usually reported as a *mean absolute error* (MAE) which is essentially the average error for the test set images. Mathematically, MAE is expressed as:

$$\text{MAE} = \frac{1}{n}\sum_{i=0}^{n-1}|y_i - \hat{y}_i| \tag{3}$$

where y_i is the predicted age and $\hat{y}_i$ is the actual age as labeled for that image. MAE is used for both age estimation and BMI estimation in this study. A 2016 survey found that FG-NET age results range from MAEs of 3.17 to 6.22 using the LOPO (leave-one-person-out) testing protocol [8]. LOPO is a variation on LOO (leave-one-out), where models are trained on every subject in the dataset except for one, and the subject who is left out rotates each time until all subjects have been left out at least once. This means that a total of 82 models will be trained, and the error on the left-out individual is averaged across all test results. A number of approaches to face-based age estimation have been taken and, before deep learning, usually involved a pipeline of feature extraction, dimensionality reduction, and regression. Since FG-NET, many larger face-based datasets have arisen and, because of its relatively small size, FG-NET is not used in this study. Gender results are usually reported as a simple binary accuracy. If the model estimates the test label correctly, then the prediction is considered to be correct and the final model accuracy is the total number of

correct predictions divided by the total number of images. Gender recognition results have exceeded 99% accuracy in several cases [9] [10], although the generalization of these models has not been studied in great extent.

With the arrival of the MORPH-II dataset in 2008, age and gender estimation could now be studied on a large scale. A larger dataset meant more training data, yielding results that were more statistically significant. MORPH-II is a collection of 55k mugshot images which were cleanly collected – every subject is standing in front of a neutral background roughly the same distance from the camera, and flash is used to illuminate the face [11].

Human Age Estimation

In contemporary AI studies, there is always a question of man vs. machine. Many vendors want to know if their algorithm can match or exceed human performance at its designed task. In 2013, Han et al. gauge human age estimation performance by crowd-sourcing estimates on both FG-NET and the PCSO (Pinellas County Sheriff's Office) dataset. PCSO is another large mugshot dataset with many of the same attributes as MORPH-II, but it is not as widely studied. Han et al. found human age estimation error on FG-NET to be 4.7 years overall, but 7.4 years on subjects over the age of 15, since younger individuals are easier to identify by age. For PCSO, the human MAE was 7.2 years since it does not contain children under the age of 15. In the study, Han et al. use a combination of SVMs (support vector machines) and extracted BIFs (biologically inspired features) to obtain an algorithmic age estimation error of 4.2 years on the MORPH-II dataset [12]. BIFs are an extension of the Gabor wavelet that are meant to emulate primate visual cortex processing, while SVMs are trainable classifiers.

Early Bias Studies

Some of the first studies of the impacts of age, gender, and race on classifier performance was performed by Ricanek and Tesafaye in 2006. They introduce the MORPH dataset which

challenged the effectiveness of contemporary face recognition and, at the time, was the largest publicly available face recognition dataset [11]. Challenging leaps in subject ages, however, were not the only thing that MORPH would be valuable for. Images were also labeled with gender, race, height, and weight, so now researchers could also begin to consider other factors that might affect face recognition performance and aging. Using the new dataset, the PCA FR algorithm was evaluated. Ricanek and Tesafaye found that younger people were the most difficult to identify, and the algorithm's accuracy rate dropped very quickly as the age span increased. The most easily identified group by far was men aged 40-49 with a span of 5 years or less and a match accuracy of 80%. They suggest that the accuracy rates reported by the FRVT2002 study were inflated due to a dataset that was too homogeneous in terms of race and gender and had very low age spans. Shortly later, the MORPH-II dataset would be released. MORPH-II will be discussed in detail in section since it is an integral part of this study.

Guo and Mu of West Virginia University [13] [14] [15]. In 2009, they compare the YGA dataset MAEs across different age ranges for their own proposed solution as well as three others [15]. YGA contains 8,000 total images and each age group contains 1,000 images – 500 male, and 500 female. They trained separate SVM models for men and women using BIFs that were modified to better extract age features. Their models far outperformed the other 3 with an MAE of 3.69 as compared to the second best solution which had an MAE of 10.08, however, females had an MAE of 3.91 while males had an MAE of 3.47. The male and female categories both have 4,000 images each, but the exact same process for both males and females led to a much higher error for females which seems to indicate a gender bias in age estimation. In [13], Guo and Mu show that race and gender have a strong impact on age estimation. They postulate that training a single general model for age estimation would cause different races and genders to have negative impacts on each other. To test this idea, they create two balanced training sets, S_1 and S_2, that have an even number of black and white people and are 25% female populated. The remaining

images are left in test set W. They evaluate their results by training a model on either S_1 or S_2, and using the rest of the dataset as test images. This train/test protocol would become a standard subsetting scheme for MORPH-II that other researchers would follow in order to compare results. Guo and Mu train age models for each race and gender category using their modified BIFs and additionally tried applying 4 different methods of dimension reduction to the BIFs. The best models were produced by BIF+OLPP (orthogonal locality preserving projections) in every category except for white females which was narrowly beaten by BIF+MFA (Marginal Fisher Analysis). As such, they use the BIF+OLPP models for each category and perform a cross race and gender analysis. In every instance, performance decreases when using a different gender or race's age estimator to make predictions. In fact, error increases by 97% when feeding test images into the opposite race and gender's age estimator. To evaluate the proposed approach as a realistic implementation, they take the BIF+OLPP features and train one more model for race and gender classification which turned out to be 96.8% accurate. After training the race and gender model, they were able to feed the test BIF+OLPP representations into it to determine which age estimator to use. The final framework yields an MAE of 4.33 which is slightly higher than the 4.28 BIF+OLPP baseline due to the 3.2% error in race+gender classification. They test their framework on the rest of the unused images with the opposite training set yielding a MORPH-II MAE of 4.45. They found race+gender accuracy to be 0.43% higher for whites than for blacks. In a follow-up paper [16], Guo and Mu attempt to find a way to estimate age, gender, and ethnicity all at once without having to develop an elaborate framework such as the one mentioned above to deal with the negative influences that race and gender differences have on age estimation. They propose that PLS (partial least squares) can be also be used for regression and not just dimension reduction. They also introduce the KPLS (kernel PLS) method which yields better race and gender classification, and a better age MAE than PLS as well as the above framework. They do not go on to evaluate the differences in age estimation performance by race and gender.

In 2010, Wang et al. detail the difficulties in classifying the gender of children [17]. Several previous studies suggested that children were difficult to identify because of their lack of distinguishable features. Wang's PCA with sequential selection algorithm yielded an 84.33% overall accuracy, but the accuracy for children aged 0-10 was only 78.1%, while the accuracy for adults aged 19-55 was 92.16% on FG-NET. Conversely, Ricanek et al. find that age estimation is easiest for children [18]. They observe that the age estimations made for FG-NET by the RPK model [19] have very low error with an MAE of 2.3 for 0-9 year olds, but extremely high error for older people aged 60-69 with an MAE of 33.15. The Ricanek et al. model balances out the error with minimum and maximum MAEs of 4.08 and 6.90 belonging to the 40-49 and 50-59 age groups respectively. The overall model MAE was 5.70. They also analyze error by race and find that age estimations are easiest for Caucasians with an MAE of 5.67, and hardest for African-Americans with an MAE of 6.90.

Deep Learning

In the mid 2000s, Fei Fei Li, who is now at Stanford University, recognized the need for a large-scale computer vision dataset. Li began assembling the dataset from WordNet, a hierarchy of words that maps the relations between words. Li used student researchers and Mechanical Turk to start gathering images depicting the words in WordNet. The resulting dataset came to be known as ImageNet, which is now one of the most important performance benchmarks in the computer vision industry [20]. The most recent release of ImageNet contains 14,197,122 images annotated with 21,841 label classes. Starting in 2010, Li began holding an annual competition known at the ImageNet challenge at the world's top computer vision conference CVPR (computer vision and pattern recognition). The dataset was extended to handle 3 computer vision tasks: *Image Classification*, *Single-Object Localization*, and *Object Detection*. Image classification simply asks the model "What is pictured in this image?". For example, a gender classification model would be shown a person's face and report "male", "female", or "other". In addition to classification, object detection or

localization involves recognizing not just what is in an image, but where it is located in the image and the boundaries of the object. This task requires bounding box labels where a human has drawn rectangles around the objects in the image to mark the coordinates where the object exists. In single-object localization, the model attempts to draw a bounding box tightly around just a single object for which the image is labeled. In object detection, the model is expected to draw a bounding box around every object that it is trained to recognize in the image. Detection accuracies are usually reported as an MAP (mean average precision), which is a calculation of the percentage that the produced labels and bounding boxes match the ground truth. For example, if an image contains a cat, and the produced bounding box covers 50% of the ground-truth bounding box that surrounds the cat, then the model is said to have a 50% MAP for that object. One type of computer vision task that is not covered by ImageNet is *Image Segmentation*. Image segmentation is usually regarded as the most difficult computer vision task because it is very difficult to label an image segmentation dataset and the images are usually processed on a pixel-by-pixel basis.

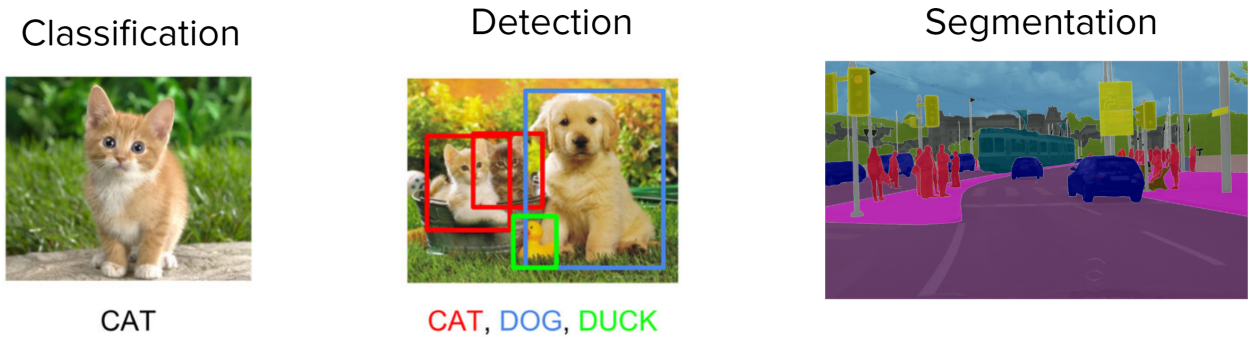

Figure 2: These are examples of common computer vision tasks. In the image segmentation example, different colors represent different object classes.

In the early days of ILSVRC (ImageNet Large Scale Vision Recognition Challenge), image classification rates were as high as 75% with the best team scoring a 28.2% error rate. That all changed in 2012 when Krizhevski et al. of the University of Toronto made a submission which scored 16.4% on ImageNet, over 11 points lower than the second best competitor. The submission, which later became known as AlexNet after Alex Krizhevski, the lead author,

was the first Deep Convolutional Neural Network (DCNN) [21]. The model took advantage of the highly parallelizable nature of GPUs, running thousands of simultaneous calculations for each layer on different threads. The powerful effectiveness of so-called deep neural networks quickly drew a lot of attention as researchers began seeing huge improvements on a wide variety of machine learning tasks. In the following years, almost all ILSVRC submissions would be DNNs, and the top-5 error rate has now dropped to nearly 3% at the time of this writing.

Age and Gender with Deep CNNs

Jumping on the success of the deep learning approach, Rothe et al. devised a solution for age estimation that allowed them to win the 2015 ChaLearn LAP (looking at people) competition. To do so, they collected one of the largest age and gender labeled datasets known as IMDB-Wiki. The IMDB-Wiki dataset is a collection of 523,051 images of celebrities crawled from the IMDB and Wikipedia websites. Each of these celebs has a reported date of birth, and every image is labeled with the year in which it was captured, so approximate age may be calculated. The dataset was cleaned by running a face detection algorithm on the raw images, and then a web-app was used to crowd-source gender labels and to verify celebrity identities. The ChaLearn LAP competition examines a different concept of age. Rather than actual age, the competition focuses on predicting apparent age. The small dataset that they released for the 2015 competition (only 4,699 images) contains labels that correspond to how old the subject appears to humans (apparent age) rather than the actual chronological age of the subject. Rothe et al. of Merantix first fine-tune the VGG-16 network on their cleaned IMDB-Wiki dataset consisting of 260,282 images. VGG-16 consists of 13 convolutional layers followed by two dense layers of size 4096 before handing the final output to the softmax layer, which gives it a total depth of 16 layers. It was designed by Simonyan and Zisserman of the Visual Geometry Group at Oxford (hence VGG) and achieved some of the top ImageNet results in 2012 and 2013 [22]. Its neural architecture components will

be discussed in greater detail in section . To fine-tune the network, Rothe et al. start with the learned ImageNet weights, and then use a low learning rate to retask VGG for age estimation using IMDB-Wiki. Then, the IMDB-Wiki models were further fine-tuned on 20 random splits of the ChaLearn dataset [23].

Fine-Tuning

Following in the footsteps of Rothe et al., Antipov et al. of Orange Labs used fine-tuning to win the 2016 ChaLearn LAP competition and achieve state-of-the-art results on MORPH-II [9]. They experimented with different CNN depths and face-recognition pretraining. Since Rothe et al. do not distribute the clean version of IMDB-Wiki, the Orange Labs group repeated the steps taken to clean the dataset. They trained their differently sized "fast_CNNs" on IMDB-Wiki but find the age and gender error to be unsatisfactory. Then, they work on fine-tuning both VGG-16 and ResNet-50. ResNet is the 2015 ILSVRC competition winner developed by Microsoft [24]. In the ResNet paper, the idea of residual connections is introduced which concatenate the output of a layer with its input, helping to stabilize the network during training. The winning ResNet models were 152 weight layers deep but are more lightweight than VGG-16. Antipov et al. use face recognition weights in VGG-16 and ResNet-50 while fine-tuning age and gender models on IMDB-Wiki. Then, they fine-tune again on MORPH-II to obtain a gender recognition accuracy of 99.4%, and allegedly an age MAE of 2.35. They found ResNet to be better for gender and VGG to be better for age. Fine-tuning on smaller datasets allows researchers to obtain very good scores on the datasets, however, this comes at the cost of loss of generalization. A network fine-tuned on the large IMDB-Wiki dataset will be broadly generalized and will have good real-world performance because of the amount of training data, but a network fine-tuned on MORPH-II will lose generalization because of the size and characteristics of the dataset. This is an example of the impacts of negative set bias. The exception to this is when a larger dataset is used for fine-tuning a model to a specific task. Bruveris et al. of Onfido fine-tune a ResNet-100 face

recognition model on a dataset of 6.8M images for the task of comparing a selfie to an ID photo. While fine-tuning has a much better accuracy than the baseline, the authors note the differences in demographic performance and report a high false match rate for people from African countries. They attempt to mitigate this effect by oversampling Africans during the fine-tuning process. They report that while they are able to get the model to perform better on Africans, accuracy is lost for people of other ethnicities. Additionally, the oversampling process seems to worsen gender bias [25]. [2]

Gender Bias in Commercial Algorithms

In 2017, Buolamwini and Gebru introduce the idea that it is much more difficult to algorithmically determine the gender of darker skinned individuals from face images. They also show that performance is usually always worse for women than men. They elucidate that many types of AI are biased, including a Word2Vec analogy generator that fills in the results of "man is to computer programmer as woman is to" with the word "homemaker" [26]. To gauge gender bias in commercial systems, Buolamwini and Gebru assemble the Pilot Parliaments Benchmark (PPB) dataset which contains 1270 images and is fairly balanced in terms of gender and Fitzpatrick skin type[3]. They use the dataset to evaluate the accuracies of 3 gender recognition algorithms produced by Microsoft, IBM, and Face++. IBM's algorithm proved to have the worst accuracy, misclassifying the gender of darker skinned people 22.4% of the time and lighter skinned people only 3.2% of the time. Microsoft's algorithm was the best overall with an accuracy of 93.7% but they still misclassified darker skinned people the most with an accuracy of only 79.2% on dark females [27].

GANs and HRFAE

Generative Adversarial Networks (GANs) are a generative technique for creating photo-realistic images that appear as if they could be part of a dataset. GANs were conceived in

[2]Oversampling only seems to shift bias from the better performing groups to the worse ones.
[3]Fitzpatrick skin types range from 1-6 where 1 is the fairest skin tone and 6 is the darkest.

2014 by Ian Goodfellow in his seminal work *Generative Adversarial Nets* [28]. GANs are essentially competing AIs that play a zero-sum game during the training process. In the game, a generator G tries to defeat a discriminator D by generating images that appear to D as if they are part of a real dataset. D will take batches of images from the real dataset, and also G's generated images, and attempt to distinguish which images have come from the generator. This min-max game is formally stated in Goodfellow's paper as:

$$\text{minmax}V(D,G) = E_{\mathbf{x}\sim p_{\text{data}}}(\mathbf{x})[\log D(\mathbf{x})] + E_{\mathbf{z}\sim p_{\mathbf{z}}}(\mathbf{z})[\log(1 - D(G(\mathbf{z})))].$$

The above formula implies that the error of the discriminator on the real data will have the logarithm of its error backpropagated through the discriminator while the error of the discriminator on the output of the generator will have the logarithm of its error backpropagated through the generator. Starting out, D will very easily detect examples created by G, and over time, G's produced examples will become more refined. The goal is usually for G to defeat D so that a realistic example which appears to the D as if the example could fit into p_{data}'s distribution. $p_z(z)$ is a random noise input vector of fixed size which is essentially the seed for the generated example. In the original paper, Goodfellow et al. demonstrate examples of generated faces, CIFAR-10 classes, and handwritten digits. Later that year, Mirza and Osindero introduce the idea of conditional GANs. *Conditional GANs* allow one to specify the desired class to be output by the generator network. In the paper, the authors use a conditional GAN to generate the specified handwritten digit (0-9). Conditional GANs have been implemented in many other studies for tasks like face generation [29], face attribute editing [30], and face aging [31].

This work uses the HRFAE GAN [32] in experiment 3 to synthetically augment dataset examples in underrepresented subpopulations. HRFAE was trained using 70,000 FFHQ [33] images and uses the Dex [23] age estimator to gauge the effectiveness with which the target

age is generated. Figure 3 shows an example usage of HRFAE GAN. An important part of the GAN's design is reconstruction loss which compares the produced image to the original. This helps to ensure that minimal edits are made to the image and nothing else about the image changes and no artifacts are present.

Figure 3: In this figure, HRFAE is used to age a woman from 25 to 55. Generated examples that are this convincing could be used to augment a dataset.

BMI Bias

Visual BMI estimation has been studied limitedly, but it has also been shown to have race and gender biases. The first such study was conducted using the MORPH-II dataset, which, in addition to age and gender, is also labeled with self-reported height and weight. The authors use the active shape model (ASM) to landmark facial features and calculate ratios such as cheekbone to jaw width and face width to lower face height. The features were then fed into three different regression models to infer estimations about BMI. The authors observed that age did not seem to have much impact on BMI estimations, however, the models performed best on white men, and worst on black women, indicating possible racial bias [34]. In 2017, Kocabey et al. use the VisualBMI dataset crawled from Reddit posts

to fine-tune deep learning models for BMI estimation. They found that human evaluation of facial adiposity was slightly better than the deep learning method. They also examined bias in the models but did not find significant bias against women or African-Americans [35]. Jiang et al. in 2019 use the FIW-BMI dataset along with MORPH-II to train deep learning models for estimating BMI. They found that fine-tuning ArcFace [36] on the FIW-BMI dataset was usually the most effective method, however, it was biased against women. ArcFace is currently one of the best performing face recognition models. The same methods applied to the MORPH-II dataset showed that it was clearly the easiest to predict white male BMI, and the hardest to predict black female BMI. When combining extracted features with PCA, VGGFace made the best predictions. Jiang et al. also perform a cross-dataset analysis using FIW-BMI and MORPH-II. They find that FIW-BMI generalizes better, producing an overall MAE of 3.48 on MORPH-II. The MORPH-II model scored 4.73 on FIW-BMI [37]. This is likely due to the lack of background noise in MORPH-II images.

Mitigation Attempts

Aside from the aforementioned age and gender bias mitigation attempts, bias mitigation has been studied on more general tasks as well. Usually the strategy is to try to "fix" the dataset. Proposed techniques such as SMOTE involved generating synthetic data from dataset minorities as a method of oversampling those minorities. That, combined with majority undersampling, showed an improvement in performance for a number of binary classification problems [38]. ADASYN was an attempt to solve data imbalance issues by learning weights for individual minority examples which affect how often those examples are to be chosen for synthetic augmentation [39]. ADASYN shows little or no improvement for some datasets and would be intractable when working with neural networks. GenSample approaches resampling dataset minorities in a similar way to SMOTE and ADASYN, however, it uses a genetic algorithm to learn weights for dataset examples [40]. GenSample did not demonstrate results that were a significant improvement over SMOTE and ADASYN,

and sometimes, the baseline even beat all three algorithms.

EVOLUTIONARY ALGORITHMS

Evolutionary algorithms (EAs) fall into the category of metaheuristic informed search algorithms. *Heuristic algorithms* attempt an action heuristically and then learn from the reaction to the attempt. A heuristic anti-virus, for example, will execute a bit of machine code in a controlled environment and observe the behavior of the code, watching for any security threats. Informed search algorithms use information to help intelligently reach a goal. Since EAs are metaheuristic informed search algorithms, they attempt an action, evaluate the fitness of that action, and then continue working towards their ultimate goal using the knowledge of the previously attempted action(s). EAs are often used to solve an optimization problem, or to optimize some process. Optimization problems are typically NP-hard and may have multiple solutions or an acceptable level of tolerance for a solution. NP-hard problem solutions lie within a search space that grows very large as the degree of the problem increases. For example, an NP-hard problem that has a time complexity of $O(n!)$ increases factorially, so a problem of size 50 could have 3.04×10^{64} possible solutions. That makes exhaustively searching the problem space computationally intractable, so typically metaheuristic informed search algorithms are used to arrive at an acceptable solution within a reasonable amount of time. One downside of evolutionary algorithms is that it is never known whether or not an optimal solution has been arrived upon without exhausting the search space, nor is it known how close the current best solution is to the true optimum. Training a neural network in itself is an optimization problem where the network is searching through a huge search space for the right combination of neuron weights which minimize error on some training set while also minimizing error on a validation set. In this thesis, a specific evolutionary algorithm is implemented to optimize bias mitigation techniques in several experiments. *Genetic Algorithms* (GAs) are a type of evolutionary algorithm that have emerged from the study of biologically inspired computing. Bio-inspired computing attempts to mimic real-world biological systems within a computer program. As such, genetic

algorithms are aptly named because they draw from the process of gene expression. GAs seek to solve optimization problems by breeding two parent genes to produce a child gene with an optional change of the gene mutating – much like what happens when two organisms procreate. Potential optimization problem solutions can be encoded in these digital genes, so the child gene represents a new solution. During the evolutionary process, the child gene is evaluated for *fitness* and then is often mutated according to a certain set of rules and probability which may be fixed or may vary over time. The fitness of a gene is evaluated using a fitness function. The fitness function uses the "chromosome" of the gene to mathematically produce a number representing the score for the gene which is either being minimized or maximized during the evolutionary process. Mutation is a method of introducing *genetic diversity* into the population of genes so that they don't get stuck in a local minimum/maximum and keep breeding the same solutions over and over again or converge to just one gene. When using a GA to optimize a problem, typically a random starting population is generated and each parent gene is evaluated for fitness. The distribution of fitness scores can usually provide a good idea of what the mean, min, and max scores look like. Then, the population is selectively bred to produce the next generation of parent genes. Certain weights are given to the parent genes based on where each gene falls with respect to the other genes' scores. For example, the algorithm might only draw from the top 75% of the genes when choosing parents to mate. Then each parent engages in *genetic crossover* with another parent where part of the genetic code is imparted in a child gene from each parent. The resulting child genes become the new population of parent genes, and typically the best gene found during the evolutionary process is recorded as the optimal solution. Populations are bred over and over again until some stopping criteria has been met. Over time, the genetic algorithm will keep finding better solutions. This strategy allows an extremely large search space to be traversed in a short amount of time yielding an optimal or near optimal solution in a matter of seconds as opposed to the potentially trillions of years that it could take to exhaustively evaluate every single possible solution. The evolutionary algorithm employed in this thesis

is discussed in more detail later in this chapter and is considered evolutionary but genetic because there is no actual crossover between candidates in the evolutionary process. Other examples of bio-inspired evolutionary algorithms are *hill-climbing*, *simulated annealing*, and *tabu search* [41].

Hill-Climbing

The hill-climbing algorithm is a simple greedy algorithm that involves iteratively searching "nearby" solutions for a better overall solution. Usually a random starting population is generated for hill-climbing and then the best solution is exploited by testing the fitness of some or all nearby solutions. If a better solution is found, then it becomes the new candidate and its nearby solutions are explored and evaluated likewise. The hill-climbing algorithm is simple, but it is rarely used in practice because of its tendency to get caught in a local minimum or maximum, and it often will not yield an optimal result [41].

Simulated Annealing

Simulated annealing is a more effective EA that is also considered greedy in nature. Simulated annealing mimics the process of metal annealing which involves gradually cooling metal from a high temperature so that the molecules fall evenly distributed into a crystalline structure which results in better structural integrity for the piece of metal. Simulated annealing involves choosing a random starting point and mutating the candidate until a simulated "temperature" is reached. Every time a candidate is mutated, its fitness is evaluated, and a distance between the fitness score is calculated for the old candidate and the mutated one. If the new candidate is better, it is always kept. If the new candidate is worse but its distance is close enough, it will be exchanged for the old candidate. If the new candidate is worse and too far away it will be discarded. Every time a mutation occurs, the "temperature" drops and the distance threshold for determining whether the new candidate is kept or discarded shrinks. This introduces genetic diversity into the search allowing better optimums to be

discovered unlike in hill-climbing. It might take several tries to find starting temperatures and decay rates that yield good results for a particular problem [41].

Tabu Search

Tabu (or "taboo") search is another evolutionary algorithm which is often used to solve optimization problems. Tabu search is similar to hill-climbing in that it picks the best candidate and explores a certain number of mutations of that candidate to find a better one. When a candidate is explored it is marked as "tabu" meaning that it cannot be re-explored until the number of tabu candidates reaches a certain size, and then the oldest tabu candidate is removed from the list, allowing it to be explored again. Every time a candidate is explored, the best mutation is chosen to be explored next whether or not it is better than the previous candidate. There are many variants of tabu search, each of which offers some sort of advantage over the other such as arriving at the optimum faster or increased diversification [41].

A Simple (yet effective) Evolutionary Algorithm

In their paper *Regularized Evolution for Image Classifier Architecture Search*, Real et al. demonstrate that evolved neural network architectures can outperform hand-crafted ones. To do this, they begin with the NASNet search space (NASNet will be described in greater detail in section) and experiment with methods of speeding up the evolutionary process. Firstly, they reduce the set of possible mutations, confining the search space. Secondly, they compare random search (RS) with reinforcement learning (RL) and a novel evolutionary algorithm – *aging evolution*. Aging evolution is a slight modification of a simple *tournament selection* algorithm. A typical implementation of a tournament selection algorithm involves these steps:

1. Generate a random starting population and evaluate the fitness of each candidate.

2. Draw a random sample from the population and select the fittest candidate.

3. Mutate the candidate and reevaluate fitness.

4. Place the new candidate in the population while removing the least fit candidate from the sample.

5. Repeat steps 2-4 until you run out of time/funding or meet some other stopping criteria.

Real et al. refer to this as *non-aging evolution*. In order to introduce genetic diversity into this process, Real et al. propose taking into account the recency with which the solution was generated, rather than just the fitness. To do so, they treat the population as a queue, and kill the oldest member of the population (the last candidate in the queue) rather than the least fit. Over time, even the most fit candidate will age out of the population, causing the algorithm to shift focus to other good candidates rather than continuously mutating very similar candidates. The aging evolution algorithm can be seen in figure 4 below. An important step I added to this process is checking the history after mutation to ensure that the candidate has not already been evaluated. For smaller combinatorial problems, duplicate candidates can result in a lack of genetic diversity and will cause the algorithm to underperform. In the Real et al. study, aging evolution reached better models faster than random searches and reinforcement learning.

A Toy Problem

For the experiments documented in this thesis, I wanted to make sure to choose the most effective EA for quickly arriving at optimal solutions without it being so greedy that it would get stuck in local minimums. The evolutionary process would involve training multiple sample neural networks, all of which would take several hours to train even on expensive equipment. To make an informed choice, several algorithms were compared on a classical optimization problem. This particular problem, the quadratic assignment problem (QAP), has existed in research since the 1960s, and has manifested in a number of different real-world scenarios. It was first postulated by engineers at MIT who, at the time, were

Aging Evolution

$population \leftarrow$ empty queue ▷ The population.
$history \leftarrow \varnothing$ ▷ Will contain all models.
while $|population| < P$ **do** ▷ Initialize population.
  $model.arch \leftarrow$ RANDOMARCHITECTURE()
  $model.accuracy \leftarrow$ TRAINANDEVAL($model.arch$)
  add $model$ to right of $population$
  add $model$ to history
end while
while $|history| < C$ **do** ▷ Evolve for C cycles.
  $sample \leftarrow \varnothing$ ▷ Parent candidates.
  while $|sample| < S$ **do**
    $candidate \leftarrow$ random element from $population$
      ▷ The element stays in the $population$.
    add $candidate$ to $sample$
  end while
  $parent \leftarrow$ highest-accuracy model in $sample$
  $child.arch \leftarrow$ MUTATE($parent.arch$)
  $child.accuracy \leftarrow$ TRAINANDEVAL($child.arch$)
  add $child$ to right of $population$
  add $child$ to $history$
  remove $dead$ from left of $population$ ▷ Oldest.
  discard $dead$
end while
return highest-accuracy model in $history$

Figure 4: The aging evolution algorithm as seen in [1].

building a computer and trying to figure out which components to place in which positions in order to minimize the amount of cable needed to bus data from component to component. This so-called "backboard wiring problem" led engineers to model the computer's design mathematically. The locations where a component could be housed were in fixed positions on the backboard (motherboard), and each component had an associated "flow" which needed to exist between itself and other components. Randomly placing the components would result in high cable usage and would drive up the cost of building the computer which, in turn, would run more slowly because of the excessive cable usage. In his paper discussing the problem, Leon Steinberg framed it as a linear programming problem and he reached a better yet sub-optimal solution. The model for the problem showed up in many other instances, such as in placing facilities with certain capabilities at specific locations in order to minimize the total number of trucking miles needed to transport items between facilities. The problem eventually became known as "the quadratic assignment problem" and has been discussed in many research papers, most of which use a heuristic search algorithm to traverse the large number of possible combinations. Formally stated, the quadratic assignment problem is the problem of assigning S facilities to S locations where D is a $S \times S$ matrix that represents the distance between each location, F is a $S \times S$ matrix that represents the flow between each facility, and P is a permutation of enumerated facility locations, in a way that minimizes the cost C in

$$C = \sum_{i=0}^{S-1} \sum_{j=0}^{S-1} D_{P_i P_j} F_{ij}. \tag{4}$$

As such, the best assignment strategy P would have the lowest cost C and would minimize the amount of cable needed to accommodate the flows F between the components at positions with distances D from each other. Notably, the traveling salesman problem is a special instance of the QAP where all of the flows are equal and the goal is simply to find the shortest path through the locations given the same start and end point. Many other variations of the QAP exist and they each address different scenarios that need to be optimized.

Algorithm Comparison

In an effort to study the effectiveness of these search algorithms, algorithm performance was compared on an instance of the quadratic assignment problem. The data for this particular problem came from a study wherein Danish engineers were consulted to help design an optimal layout for a German hospital that was built in 1972 [42]. The floor plan for the hospital was already complete, but the problem was in planning which facilities should go in specific hospital wings in order to minimize patient transport time. In this instance, there were 32 facilities to be placed at 32 locations, and the flows between facilities were estimated by the hospital's doctors on a scale of 0-4. It is common in the QAP for there to exist a number of facilities with zero flow between each other, and if several of the flows are the same, there could be many optimal solutions. Given a size of 32 facilities and locations, there are 32! possible facility assignments or 2.63×10^{35}. This makes a metaheuristic search approach desirable as an exhaustive search would be computationally intractable. Random search, tabu search, aging evolution, and non-aging evolution were all used to generate one million solutions to the QAP. [4] Random search involves randomly generating solutions and evaluating them without using any informed search strategy. An analysis of the random search indicated that the solution distribution was relatively normal with a mean solution of 137,100 as can be seen in figure 5. The best solution found during the random search was 115,210 – far higher than the true minimum. Tabu search, aging evolution, and non-aging evolution were a success, but the greediness and lack of genetic diversity of non-aging evolution caused it to lag behind. The solutions generated by tabu search and aging evolution were very similar, so a side-by-side comparison was conducted to see which algorithm arrived at the true optimum the most often. Also seen in figure 5, aging evolution was the clear winner, probably because tabu search will end up marking a one-off solution as "taboo" until it is removed from the tabu list. Krarup and Pruzan's final solution was only one swap

[4] Hill-climbing and simulated annealing were not compared because, while they quickly get to good solutions, they are too greedy and rarely arrive at the optimum.

away from optimum [42]. The optimal solution for this instance of the problem was not proven until the year 2000 when a branch-and-bound algorithm for the QAP was used to exhaustively eliminate the possibility that a better solution could exist. The algorithm took nearly 100 days to run [43]. According to Moore's law, a computer of the same price today could complete the algorithm in around 2 hours.

Figure 5: Search algorithm results for the Kra32 QAP. Smaller is better. In the top graph, bins are normalized by dividing each bin count by the total number of runs.

Proposed in 1989 as a method of digital image recognition [44], convolutional neural networks have become the basis of nearly all deep learning algorithms. In the context of image processing, convolution is a matrix operation that involves the use of a kernel (often referred to as a filter) to enhance qualities of the image. Mathematically, a 2-dimensional convolution can be expressed as:

$$\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\mathbf{x}_{(m-i)(n-j)}\mathbf{y}_{(i+1)(j+1)} \tag{5}$$

where m is width and n is height. For example given that:

$$\mathbf{x} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \mathbf{y} = \begin{bmatrix} 152 & 128 & 201 \\ 94 & 115 & 167 \\ 112 & 137 & 122 \end{bmatrix} \tag{6}$$

where x is a convolutional filter and y is a small monochrome image, formula 5 results in the arithmetic of:

$$\mathbf{x}_{33}\mathbf{y}_{11} + \mathbf{x}_{32}\mathbf{y}_{12} + \mathbf{x}_{31}\mathbf{y}_{13} + \mathbf{x}_{23}\mathbf{y}_{21} + \mathbf{x}_{22}\mathbf{y}_{22} \ldots + \mathbf{x}_{11}\mathbf{y}_{33} = 535. \tag{7}$$

Filters are usually small (as in 3x3 or 5x5) because convolution is an expensive process, especially when it is used repeatedly throughout a network. Convolution has traditionally been used in digital image processing as a method of edge detection or sharpening/blurring an image. Figure 6 below is an example of convolution being applied to an image as a method of edge detection. In the image, Prewitt's gradient is used to enhance the edges of objects, making them more easily separable.

In a convolutional neural network, filters are not simple 1s and 0s – they are learned floating point values. Each value in a filter is a simulated neuron as depicted in figure 1, so while a CNN is being trained, it is learning the convolutional filters that help it

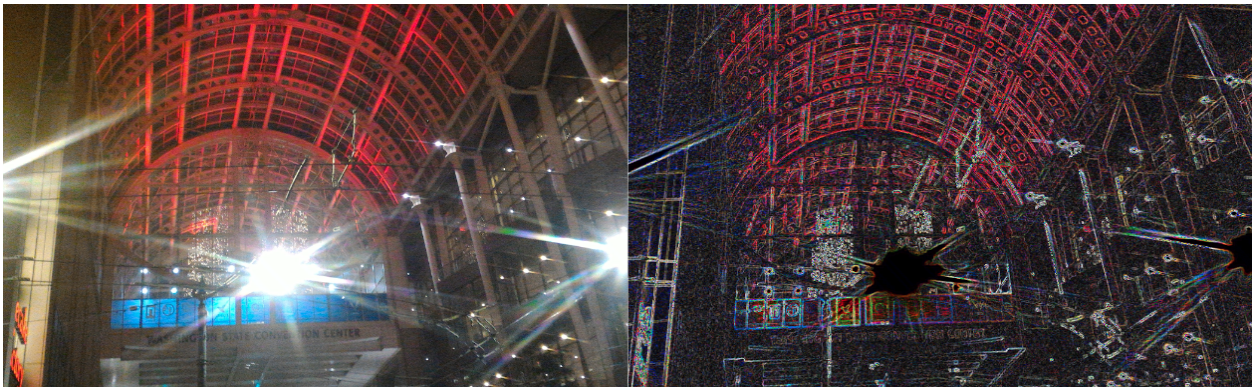

Figure 6: This image has been convolved using Prewitt's gradient for the purposes of edge detection.

best extract features that aid in its designated classification task. Typical modern neural network architectures consist of many hidden intermediary layers of convolution and pooling to extract features and reduce feature size. The copious layers led them to be dubbed "deep" neural networks. Each layer of a DNN relies on the output of the previous layer in order to begin processing extracted features, so training a DNN is a very time-consuming process. It is noteworthy that matrix operations like convolution, which make up the vast majority of a DNNs trainable parameters, are highly parallelizable on computer hardware due to the numerous multiplication and addition operations. This is part of what fueled the deep learning revolution in 2012. Prior to 2012, smaller neural networks had been trained with some success for face-based image processing tasks [45] [46], but with the evolution of GPU (graphics processing unit) technology, it was soon realized that architectures could grow in size to increase memorization and generalization. As such, deep CNNs quickly took the scene since they were able to leverage datasets for a better accuracy overall, which could keep increasing as more data were seen. Prior strategies, usually involving some form of dimensionality reduction and linear classification, could not keep improving and adapting the way that DNNs could. In addition to convolution, pooling is a common building block of convolutional neural networks. Several filters might be used in a single layer, which causes the size of the extracted features in memory to grow very quickly. As such, techniques like pooling are applied to reduce feature size. A typical 2-dimensional pooling filter might have

a width and height of 2x2. Unlike convolutional filters, however, pooling layers do not have any weights or biases to apply. Max pooling can be thought of as a window that slides along feature layers picking the max value in each window. Looking back figure 6, if we were to apply 2x2 max pooling to $\mathbf{y}$ it would result in:

$$\begin{bmatrix} 152 & 201 \\ 137 & 167 \end{bmatrix} \tag{8}$$

because those are the max values in each 2x2 window. Notice that it has reduced the size of the image from 9 values to 4. There is another common pooling method called global average pooling which is usually used when there are a large number of smaller features. If we apply global average pooling to $\mathbf{y}$, the result is 136.44 because that is simply the average of all the values. With both convolution and pooling, there is the concept of stride. Stride is essentially the number of pixels to skip between operations. For example, a 2x2 max pooling filter with a stride of 2 will skip every other "window" of pooling, effectively halving the output size. This also keeps some values from becoming too prevalent. In the above example, if we swap 201 with 115, the output becomes:

$$\begin{bmatrix} 201 & 201 \\ 201 & 201 \end{bmatrix}, \tag{9}$$

so a stride of 2 would limit the value of 201 so that it only appears twice rather than four times. There are a few more concepts necessary for understanding the inner-workings of modern neural nets. Two basic concepts are epochs and batches. An epoch is simply one pass through the entire training set. DNNs are usually shown a training set for hundreds or even thousands of epochs before no more improvement is seen. Improvement is usually gauged on a validation set, which is a set of unseen examples upon which the model's accuracy, or lack thereof, is evaluated after every epoch. A third part of the dataset, the test set, consists of held-out data which is not used at all until after the model is finished training. The test set

gives one a realistic sense of how accurate the model is and helps make sure that the model is not overfit to the validation set. A batch is a small chunk of the training set – maybe 32 or 64 images. During an epoch, data are fed through the network in batches and all images must reach the top of the network before any error is backpropagated through the network. Backpropagation is an algorithm that is essentially the way that neural networks learn from data. At the top of the network, error is calculated using the ground-truth labels of the data in the batch using a loss function. Many loss functions have been proposed, such as categorical crossentropy loss and mean squared error loss. The error is typically modified by a learning rate which can be specified manually, but is usually dictated by an adaptive gradient descent optimizer. There are many other tweaks and techniques that exist inside of a network such as batch normalization and dropout regularization. Batch normalization standardizes feature data between layers, which helps the network stabilize more quickly and reach better accuracies faster, but it comes at the cost of greater computational complexity. Dropout regularization specifies a certain probability for neurons to randomly "dropout" of the network during both the feedforward process and backpropagation. This ensures that neurons are not becoming codependent on one another. Other newer techniques include residual connections and skip connections. Residual connections add the features output by the previous layer to the features output by the current layer, which helps stabilize the network more quickly. Skip connections are similar except that they may skip many layers and they concatenate the old features to the new ones rather than adding them. There are many other DNN building blocks such as dense layers, separable convolution, and convolutional transpose, but discussing them is not necessary to this thesis.

THE DATASETS

For this study, five different datasets are used to train and evaluate age, gender, and BMI estimators. Every dataset used in this thesis has been studied in the academic community. The specific datasets used are referred to by the common names of: MORPH-II, IMDB, Wiki, MORPH-IV, and PPB. Each dataset has different characteristics, so each can be employed in different ways. Before any models are trained, the datasets are cleaned and preprocessed in a uniform manner. MORPH-II was the starting point for the preprocessing since it is the most widely studied of the five datasets. First, labels were examined for inconsistencies such as gender mismatches for the same individual, or inconsistencies in birthdates. These labels were corrected manually. Then, all images were resized to 200x240 because most images in the MORPH-II dataset are that size or larger. A face detector was used to calculate the mean face width and height [5]. Only faces larger than 50x50 were considered. Mean width face W was found to be 113.48px while mean face height H was 130.86px. Next, the images were surrounded with a border using `cv2.BORDER_REPLICATE`, and faces were centered and cropped from the image using algorithm 1 below. In algorithm 1, i_w and i_h are the target width and height of 200x240 respectively.

The IMDB-Wiki dataset is distributed in raw form so it must be cleaned. To do so, faces were cropped from each image using algorithm 1, where only one face is detected in the image with a face score of higher than 60%. If only one face is detected in the image, then it is likely that the face corresponds with the provided labels. After that, the dataset is cleaned by identifying all images where the age or gender of the subject is vastly different from the provided labels. Pretrained age and gender models from [48] are used to estimate age and gender to identify potentially mislabeled images. Then, all images which have been identified as mislabeled are manually verified, and images that are not clearly mislabeled are returned to the dataset. MORPH-IV is cleaned in a similar manner as MORPH-II. MORPH-IV is also

[5]The face detector is a Mobilenet-V2 model that was trained on the WIDER-FACE dataset [47]. It is available at https://github.com/yeephycho/tensorflow-face-detection.

Algorithm 1: Crop face to mean face size

Result: A face-cropped image.

`/* the bounding box corners below are returned by the face detector */`

x_min; y_min

x_max; y_max

center_x = $\frac{x\_min+x\_max}{2}$

center_y = $\frac{y\_min+y\_max}{2}$

face_width = w = x_max - x_min

face_height = h = y_max - y_min

`/* W and H are defined above. */`

if $\frac{w}{h} > \frac{H}{W}$ **then**

  resize_ratio = rr = W/w

else

  resize_ratio = rr = H/h

end

center_x = rr * center_x

center_y = rr * center_y

`/* resize the image in the x and y directions by resize_ratio */`

img.resize(rr, rr)

`/* crop an image of size` $i_w \times i_h$ `around the center of the face */`

img.crop(center_x, center_y, (i_w, i_h))

a collection of mugshots, and is, as could be expected, an extension of MORPH-II, except that it is more balanced by age, gender, and race. MORPH-IV additionally contains height and weight labels, so it is used in the BMI bias experiment in section . Some MORPH-IV labels had to be adjusted in instances where there were improbable heights, weights, or ages. Additionally, some images do not have height or weight labels, so they could not be used. PPB was also cleaned using the same face detection algorithm, and every face was visually verified. Table 1 shows the cleaned dataset sizes along with the labels that are used during this thesis. Additionally, the female ratio is shown because it's a very important part of the trained algorithms.

Table 1: Dataset Sizes and Labels

Name	Labels	Size	Female Ratio
MORPH-II	Age, Race, Gender	55,038	15%
IMDB	Age, Gender	131,091	43%
Wiki	Age, Gender	35,169	24%
PPB	Gender	1,166	45%
MORPH-IV	Age, Race, Gender, Height, Weight	386,739	34%

Another important feature of the datasets is the age distribution of each dataset which can be seen in figure 7 below.

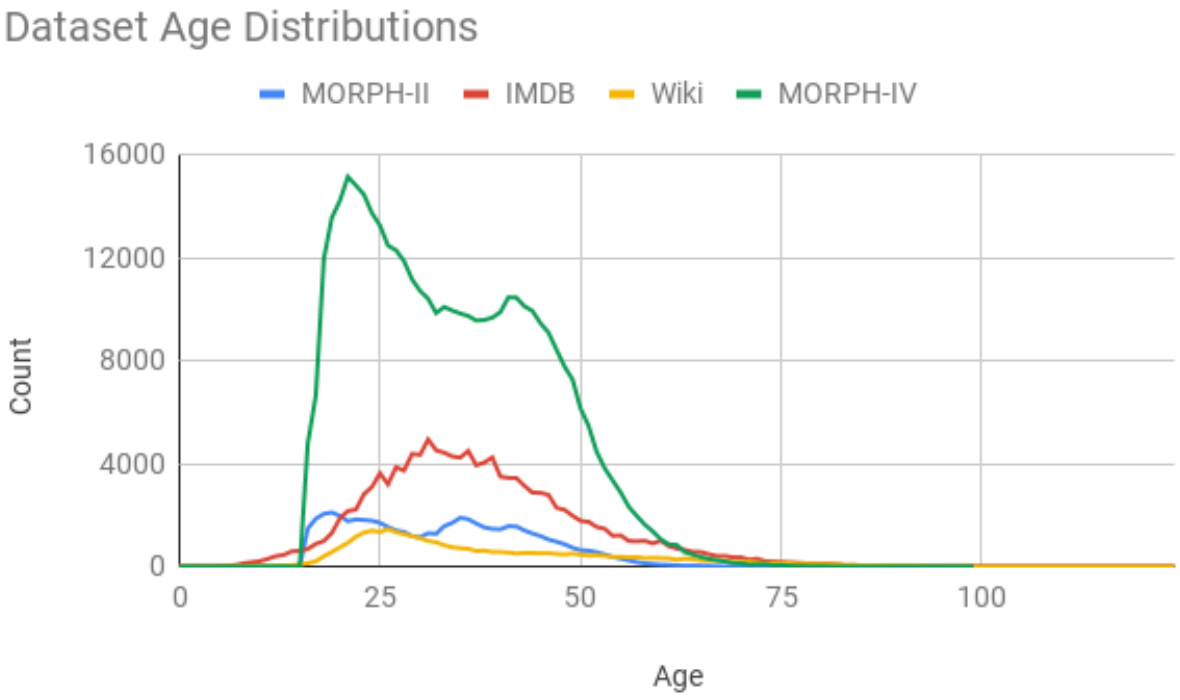

Figure 7: This histogram shows the number of individuals at each age in each age-labeled dataset. Wiki has the most even distribution but is also the smallest and most challenging dataset.

PROPOSED EXPERIMENTS

As aforementioned, this thesis attempts to address AI bias by using novel techniques to mitigate bias. To do so, three different experiments are proposed that focus on analyzing bias and reducing it. While bias is often not the topic of contemporary deep learning studies, techniques from these studies can be utilized to design a system that is more fair. Three strategies were chosen to take-on the AI bias issue: data augmentation, dataset oversampling, and synthetically generated data.

Data augmentation has proven to generalize models, while decreasing error, by manufacturing more data upon which a model can be trained [49] [50] [51] [52] [53]. Hand-picked data augmentation strategies will sometimes decrease error, but can also lead to an increase in error. Thus, for this work, data augmentation policies are adopted from [54], and an evolutionary algorithm is used to optimize the policies. The main difference between this work and [54] is that the policies are evolved against a challenge set. The challenge set contains demographic groups which are known to be difficult to make estimations for. By evolving policies against the challenge set, it is presumed that the policies will learn how to augment data in a way that best reduces error for challenging groups. The application of data augmentation policies has been shown to reduce error by a significant amount, but this can actually worsen the bias ingrained in a model by decreasing error for dataset majorities and increasing error for dataset minorities. The use of the challenge set in the evolutionary process should mitigate this effect.

Oversampling has long been considered as a method of "fixing" a dataset so that a machine learning model does not become biased against minorities. This work proposes using oversampling to mitigate bias by drawing more frequently from groups that are difficult to identify. Rather than assigning arbitrary sampling rates to each group, an evolutionary process is used to learn weights for dataset sampling. Oversampling is also combined with the data augmentation policies learned in the first experiment to see if it further reduces

bias.

Synthetic data has been addressed in techniques like SMOTE [38] and ADASYN [39], however, this work takes a relatively new approach to the idea of training on synthetic data. Since GANs have recently proven to be capable of generating photorealistic images, it is proposed that generated images could be used as training data. This third experiment addresses bias by augmenting dataset minorities using GAN generated images. Specifically HRFAE GAN [32] is used to fill out the dataset in sparse demographic groups. The next three chapters cover the details of these proposed experiments as well as their results.

EVOLVING A DATA AUGMENTATION POLICY

This experiment involves the use of data augmentation to mitigate bias in age and gender estimators. The MORPH-II, Wiki, and IMDB datasets are combined to produce an age and gender labeled dataset of 221,298 images. Due to the necessary level of specificity, set notation is used to distinguish parts of this dataset. The combination of the three datasets S is used to produce a training set S_{tr} which comprises 50% of the dataset. The validation and test sets S_{va} and S_{te} comprise 25% each. This scheme was chosen over the usual 80/10/10 train/validate/test splits for two reasons. Firstly, enough validation data needed to be present in order to create the challenge set (section). Secondly, this study was not designed to achieve the lowest error rates, but to show how a limited amount of data can be augmented to mitigate the effects of algorithm bias. Subgroups of the dataset can be defined by the image labels. Every subset is a proper subset of S and $|S| = 221,298$. Let G_g be the set of all gender labeled images where $g \in \{m, f\}$. G_m is all male labeled images and G_f is all female labeled images. Not all dataset images are gender labeled so $|S - (G_m \cup G_f)| = 2,812$. D_d is the set of all age labeled images where $d \in \{x \in Z | 0 \leq x < 10\}$ and ages for each decade are in the range $[10d, 10(d+1)-1]$, so D_2 would consist of all 20-29 year-old subjects. The final subset is R_r which is the set of all race labeled images where $r \in \{b, w, o\}$ (black, white, and other), and $R \subseteq M$. The "other" label refers to all MORPH-II images not labeled black or white so $R_o = R - (R_b \cup R_w)$. In addition to racial bias, error is observed based upon face illumination. A mean pixel value μ is calculated for the region of the face that exists in 80% of the mean bounding box (WxH). This tighter face crop helps to exclude background and hair. In set B, or face brightness, B_0 is the set of all of the darkest faces with $0 \leq \mu < 85$, B_1 is the set of all moderate faces with $85 \leq \mu \leq 170$, and B_2 is the set of all light faces with $170 < \mu \leq 255$. See figure 8 for examples.

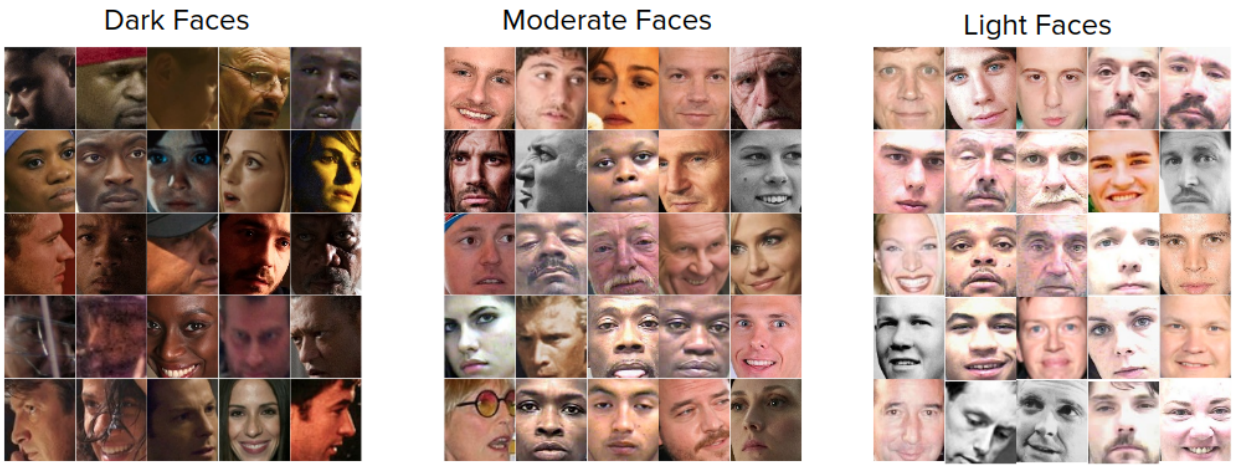

Figure 8: Separating the dataset by face brightness illustrates the general trend in accuracy by skin tone and also tests the model's performance in poorly-lit conditions.

The Model

The Inception-ResNet-v2 DCNN model was chosen for this project because of its strong ImageNet challenge [20] performance and its relatively lightweight design. Inception-ResNet-v2 combines Google's Inception modules [49] with Microsoft's residual connections [24] to yield a 572 weight-layer neural network. Its architecture consists of an input stem which downsamples the image three times before using several stacks of Inception A, B, and C modules, separated by Reduction modules, to extract features before passing them to the final global average pooling and softmax layers. With an age output layer of 100 nodes, and a gender output layer of 2 nodes, both models consist of about 55 million weights and biases. Age models use MAE (mean absolute error) loss while gender models use binary crossentropy loss. Binary crossentropy was chosen for the gender model because training set loss gets very low and a mislabeled or non-cisgender image could disrupt the weights if a higher loss penalty is enforced. All models were trained with the adadelta optimizer [55] because it eliminated the need to tune the learning rate and decay. All input is standardized to the training set mean and standard deviation after data is augmented (in the non-baseline models) but directly before being fed into the network. DLDAE (dynamic label distribution age encoding) [48] is used to encode and decode age labels, and age error is reported as

an MAE. As such, all age results are an average error in years. Gender recognition (GR) is interpreted as a binary score. If the gender network output layer node 0 has a value of greater than .5, then it has rendered a prediction of female, otherwise it has predicted male. Gender recognition accuracy is reported in terms of binary accuracy and also as an F_1 score. The F_1 score is the harmonic average of precision and recall, so it is used to more accurately observe the change in model bias. If tp is a true positive, tn is a true negative, fp is a false positive, and fn is a false negative, then precision $P = \frac{tp}{tp+fp}$ and recall $C = \frac{tp}{tp+fn}$. An F_1 score is then calculated via equation 10.

$$F_1 = 2\frac{PC}{P+C} \tag{10}$$

Baseline Examination

To identify sources of bias, separate models are trained on $S_{tr} \cap D$ for age and $S_{tr} \cap G$ for gender. During training, a randomly selected $\frac{1}{4}$ of S_{va} is used as validation data to save the top 5 models in terms of lowest validation loss. Training is stopped when the model goes for 10 or more epochs with no improvement. All results are reported only on S_{te}. S_{va} results are ignored although they are very similar to S_{te} which indicates a lack of validation set overfitting. The cardinality of each subgroup in the test set is shown in table 6. As can be seen, some subgroups are severely underrepresented in the dataset and this tends to be the reason for high error. There is an inverse correlation of -0.3222 between S_{te} subgroup count and age error. An age error standard deviation σ is also calculated for each subgroup which shows that estimations can vary wildly for underrepresented subgroups. The S_{te} MAE overall was 4.62 years – 4.36 for males and 5.13 for females. The results in tables 2, 3, and 5 help identify the greatest sources of age and gender error by subpopulation. In almost every category the baseline models perform better on men than women. This is to be expected since women only comprise 33% of the dataset. Both age and gender models have a difficult time on children and the elderly. The gender recognition model tends to identify

boys and young men as women but the reverse is true for the elderly. Faces that are more poorly illuminated are also not as easily identifiable by both the age and gender models. The results for each distinct dataset vary largely as well. The error for MORPH-II is much lower than Wiki and IMDB for a few different reasons. MORPH-II contains no children or elderly past the age of 77, and it was not captured in the wild. Every image is a frontal face image that was taken with camera flash in-front of a neutral background. Wiki is the most challenging dataset because of its lower image count and more frequent facial obstructions. Wiki dataset images tend to be captured by amateur photographers whereas most IMDB photos are taken from movies or shows. It should be noted that the low counts in some categories cause a high standard deviation for error depending on the dataset split.

Table 2: Baseline Results by Decade

	$S_{te} \cap G_m$ Male Age MAE — σ	$S_{te} \cap G_f$ Female Age MAE — σ	$S_{te} \cap G_m$ Male GR Error	$S_{te} \cap G_f$ Female GR Error
$\mathbf{D_0}$	9.24 — 12.35	11.12 — 13.16	27.66%	5.80%
$\mathbf{D_1}$	3.85 — 6.14	6.43 — 8.35	5.51%	2.83%
$\mathbf{D_2}$	3.79 — 5.27	4.18 — 5.50	1.47%	1.50%
$\mathbf{D_3}$	3.70 — 4.98	4.03 — 5.24	1.24%	1.15%
$\mathbf{D_4}$	4.36 — 5.77	5.96 — 7.52	0.99%	1.67%
$\mathbf{D_5}$	5.50 — 7.19	8.26 — 10.40	0.75%	3.06%
$\mathbf{D_6}$	6.24 — 8.41	9.34 — 12.38	0.60%	4.14%
$\mathbf{D_7}$	8.53 — 10.54	12.77 — 15.62	0.73%	7.45%
$\mathbf{D_8}$	13.14 — 14.69	15.24 — 17.53	0.00%	4.35%
$\mathbf{D_9}$	16.94 — 18.17	26.02 — 29.78	0.00%	28.57%

Data Augmentation

In this instance, the goal of mitigating bias in a trained AI model involves targeting subgroups with high error. Ideally, the error for these subgroups can be decreased without increasing the error for other subgroups. A traditional approach to reducing model bias in statistical classifiers like SVMs is to partition the dataset into subgroups that are more balanced [16]. Deep neural networks benefit greatly from being trained on large datasets

Table 3: Baseline Results by Race and Illumination

	$S_{te} \cap G_m$ Male Age MAE — σ	$S_{te} \cap G_f$ Female Age MAE — σ	$S_{te} \cap G_m$ Male GR Error	$S_{te} \cap G_f$ Female GR Error
$\mathbf{R_b}$	2.69 — 3.53	3.64 — 4.65	0.59%	1.14%
$\mathbf{R_w}$	2.69 — 3.52	3.46 — 4.54	0.00%	1.52%
$\mathbf{R_o}$	2.78 — 3.64	3.61 — 4.31	0.00%	0.00%
$\mathbf{B_0}$	5.58 — 7.41	6.10 — 8.16	2.57%	3.55%
$\mathbf{B_1}$	4.22 — 5.84	5.06 — 6.86	1.32%	1.52%
$\mathbf{B_2}$	3.27 — 4.69	4.16 — 5.62	0.95%	2.03%

Table 4: Baseline Results by Dataset

	$S_{te} \cap G_m$ Male Age MAE — σ	$S_{te} \cap G_f$ Female Age MAE — σ	$S_{te} \cap G_m$ Male GR Error	$S_{te} \cap G_f$ Female GR Error
M	2.70 — 3.53	3.58 — 4.61	0.52%	0.96%
I	5.07 — 6.80	5.89 — 7.10	1.96%	1.25%
W	5.34 — 7.18	5.25 — 7.88	1.68%	5.99%

Table 5: Baseline Gender Scores

	$S_{te} \cap G_m$ Male	$S_{te} \cap G_f$ Female
Precision	98.55%	98.23%
Recall	99.10%	97.14%
$\mathbf{F_1}$	98.82%	97.68%

Table 6: Test Set Cardinalities

	$S_{te} \cap G_m$ Male	$S_{te} \cap G_f$ Female
	Age — Gender	Age — Gender
$\mathbf{D_0}$	45 — 47	74 — 69
$\mathbf{D_1}$	2443 — 2396	1380 — 1345
$\mathbf{D_2}$	8588 — 8630	6000 — 6123
$\mathbf{D_3}$	10370 — 10427	6029 — 5988
$\mathbf{D_4}$	8352 — 8289	3039 — 3002
$\mathbf{D_5}$	3926 — 3984	1048 — 980
$\mathbf{D_6}$	1701 — 1667	456 — 483
$\mathbf{D_7}$	706 — 686	176 — 188
$\mathbf{D_8}$	193 — 199	75 — 69
$\mathbf{D_9}$	31 — 27	10 — 14
$\mathbf{R_b}$	9222 — 9276	1413 — 1401
$\mathbf{R_w}$	1937 — 1929	653 — 656
$\mathbf{R_o}$	452 — 430	31 — 36
$\mathbf{B_0}$	5495 — 4949	2310 — 1859
$\mathbf{B_1}$	28321 — 28127	14707 — 14630
$\mathbf{B_2}$	2539 — 3276	1270 — 1772
$\mathbf{M}$	11611 — 11635	2097 — 2093
$\mathbf{I}$	18218 — 18186	14569 — 14014
$\mathbf{W}$	6526 — 6531	2304 — 2154

given their ability to keep generalizing as more examples are seen. As such, a reduction in some of the larger male populations such as D_2 though D_4 would not result in more robust filters for women, but only less robust filters for men. Since augmenting data has been identified as an efficacious method for virtually increasing the size of a dataset, it can be reasoned that data augmentation techniques could be learned that most effectively augment data for the targeted subgroups. For this experiment, the data augmentation policies from [54] are adopted with some minor improvements. The data augmentations policies in [54] consist of 5 subpolicies with 2 image manipulation techniques per subpolicy. During training, when an image is loaded, a subpolicy is randomly chosen, and its image manipulation techniques are applied. This work applies the evolutionary strategy from [1], rather than the original reinforcement learning technique used in [54], because aging evolution proved to more quickly arrive at near-optimal solutions in a neural architecture search space. The evolutionary algorithm also allows for changes in the number of configurable parameters per data augmentation technique. Rather than using evenly spaced settings for each technique in the policy, this work chooses a set of reasonable defaults to narrow the valid search space. The boundaries and increments for these defaults are based on the results of [54] and also our own prior experimentation with manipulating settings. For example, posterizing image colors down to 3 bits would eliminate most of the features from the image and would not make for good training data, so the posterization technique is limited to 4 bits or higher. The evolutionary algorithm, as opposed to the reinforcement learning strategy, also allows for the expansion and contraction of techniques within a subpolicy. So while the data augmentation policy (DAP) always maintains 5 subpolicies, the number of image manipulation techniques applied by each subpolicy can change. This work also introduces the concept of top-level policies. Some data augmentation techniques have proven to be effective in many other works, so they are considered with a learned probability to be applied to each image after applying a subpolicy. The top-level policies selected for this work are horizontal flipping, mixup [56], cutout [53], and DLDAE. DLDAE is not a form of data augmentation but

it is implemented as a top-level policy in order to optimize effective hyperparameters for it. Similar to [54], every data augmentation technique, including the top-level techniques, are applied with a probability p where $p \in \{0.1x | x \in Z, 0 < x \leq 10\}$ except where otherwise specified in table 7. Some of the image manipulation techniques from [54], such as pixel-wise inversion, were not considered because, though they might yield valuable information when considering images of house numbers, they would not improve the quality of a face-image model. Two new data augmentation techniques are introduced: an improvement to random crop resampling, and a noise robustness technique called random Gaussian tinting. If these techniques did not prove effective then they would be eliminated from subpolicies during the evolutionary process.

Random Crop Resampling

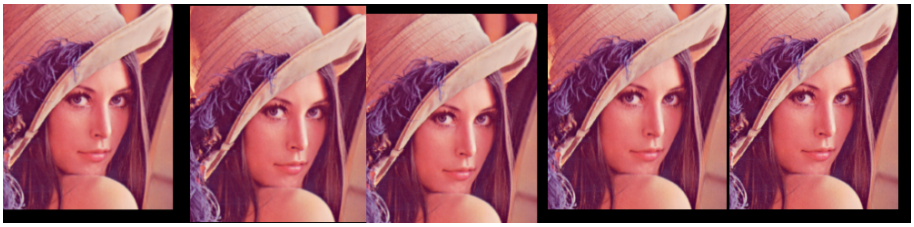

Figure 9: Random cropping with $a = 15$.

Earlier in the deep learning revolution, image resampling was commonly used as a form of data augmentation [51] [49]. 12-crop resampling, for example, involved taking a crop from each corner of an image, the middle of the image, and resizing the full image down to the target size. Each of these 6 crops would then be flipped horizontally and then all 12 unique images would be fed into the network as training data. Random crop resampling was devised as a way to handle minor discrepancies in image center, scale, and translation all at once. A slight change in a face's scale should not cause a vastly different age prediction. When random crop resampling is applied, a black border of a width learned using the reasonable defaults seen in table 7 encompasses the image. Then, the image is cropped with a random center down to a size that is proportional to, but not necessarily equal to, the input size of

the network. Finally, the image is resized to fit the network.

Algorithm 2: Random Crop Resampling

Result: A randomly cropped image.

def randomly_crop(img, border_width)**:**

img.add_border(border_width, color=black)

`/* randomly choose a top-left corner for the crop that will cause it to be larger than or equal to the target size */`

x_min = randint(0, border_width * 2)

y_min = randint(0, border_width * 2)

`/* ratios to` i_w `and` i_h `of potential crop size from the top-left corner */`

remaining_w_ratio = wr = $\frac{(i_w + border\_width * 2) - x\_min}{i_w}$

remaining_h_ratio = hr = $\frac{(i_h + border\_width * 2) - y\_min}{i_h}$

width_is_min = wr ¡ hr ? true : false

if width_is_min **then**

  crop_w = i_w + randint(0, i_w + border_width * 2 - x_min)

  crop_h = crop_w * $\frac{i_h}{i_w}$

else

  crop_h = i_h + randint(0, i_h + border_width * 2 - y_min)

  crop_w = crop_h * $\frac{i_w}{i_h}$

end

x_max = crop_w + x_min; y_max = crop_h + y_min

img.crop(x_min, y_min, x_max, y_max)

img.resize(i_w, i_h)

return img

Random Gaussian Tinting

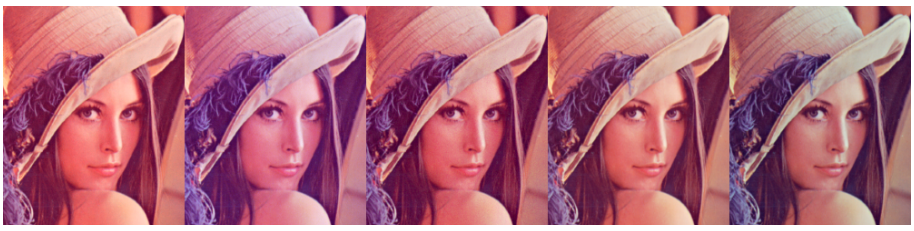

Figure 10: Random Gaussian Tinting with $a = 30$ and $b = 5$.

In addition to random crop resampling, random Gaussian tinting is introduced as a method of increasing model robustness. A small amount of noise on an image should not greatly effect the output of a network; nor should a slight variation in the tint of the image. Random Gaussian tinting works by choosing a subset of the R, G, and B channels, and adding or subtracting a layer of Gaussian noise from those channels. The mean of the

Gaussian distribution from which noise is drawn may be shifted by its learned a value, and the sampled noise will have a standard deviation of b (see table 7). For example, if $a = 50$ and $b = 5$, then the average color channel shift for the selected channel will be $\pm$50px and the noise standard deviation will be 5px. In this way, if the blue and green channels are selected to be modified, and the sampled noise is added to both layers, then the resultant image will have a teal tint to it and will appear to be of slightly lower quality due to the noise.

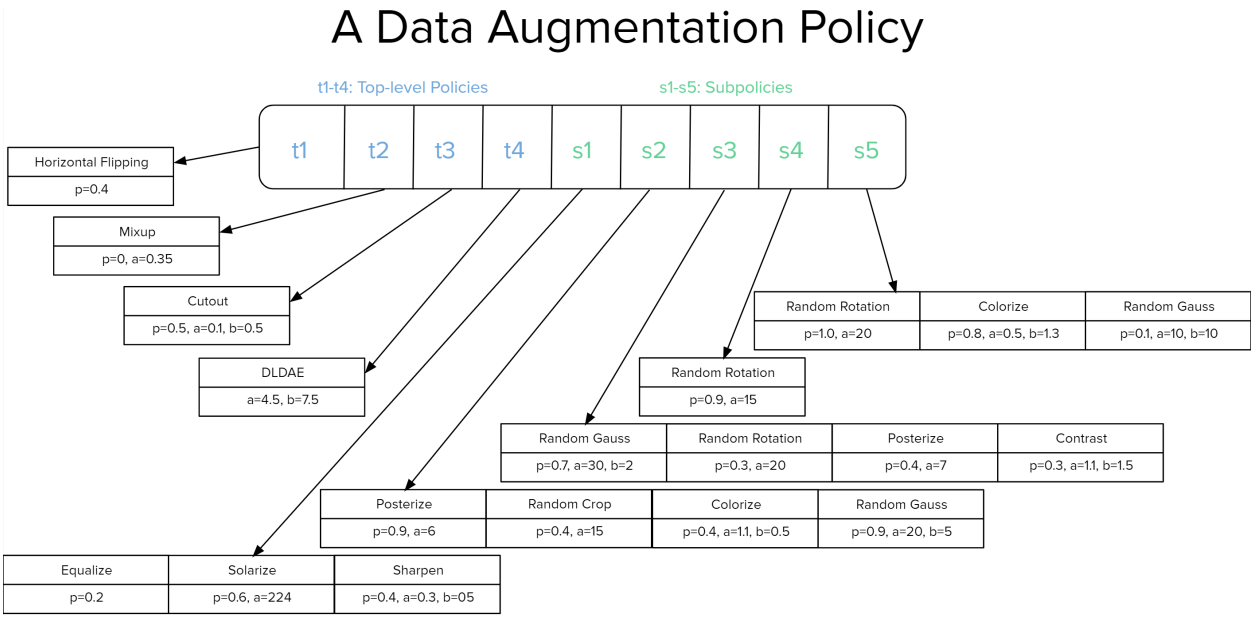

Figure 11: This figure represents a data augmentation policy proposed by this work. The top policy learned during evolution is depicted here.

The Challenge Set

The results of the baseline tests are used to partition a challenge set that contains subgroups that are high in error. As noted earlier, there is a clear gender bias in the baseline models. Children and the elderly tend to be particularly difficult to identify in terms of age and gender, and darker faces also pose a challenge. As such, the challenge set is comprised of validation set images that fit these difficult categories. Only images from S_{va} are used. During the evolutionary process, the challenge set is used as validation data to evaluate the

Table 7: Data Augmentation Techniques

Technique	**Reasonable Defaults**	**Description**
Color	$a \in \{.3, .5, .7, .9, 1.1, 1.3, 1.5, 1.7\}$, $b \in \{.3, .5, .7, .9, 1.1, 1.3, 1.5, 1.7\}$	Adjust the color balance of the image by a randomly chosen amount between a and b.
Contrast	$a \in \{.3, .5, .7, .9, 1.1, 1.3, 1.5, 1.7\}$, $b \in \{.3, .5, .7, .9, 1.1, 1.3, 1.5, 1.7\}$	Adjust the contrast of the image by a randomly chosen amount between a and b.
Equalization	-	Equalize the histogram of the image.
Posterization	$a \in \{4, 5, 6, 7\}$	Reduce each pixel value to a bits.
Random Cropping	$a \in \{5, 10, 15, 20, 30, 40, 50\}$	See paragraph .
Random Gaussian Tinting	$a \in \{10, 20, 30, 40, 50, 60, 70\}$, $b \in \{2, 5, 10, 15, 20, 25\}$	See paragraph .
Random Rotation	$a \in \{5, 10, 15, 20, 25, 30, 35\}$	Rotate the image by a number of degrees between a and $-a$.
Sharpening/Blurring	$a \in \{.3, .5, .7, .9, 1.1, 1.3, 1.5, 1.7\}$, $b \in \{.3, .5, .7, .9, 1.1, 1.3, 1.5, 1.7\}$	Sharpen or blur the image by an randomly chosen amount between a and b. Image is sharpened if the value is greater than 1 or blurred if the value is less than 1.
Solarization	$a \in \{192, 208, 224, 240\}$	Invert the pixel values that are higher than a plus a random integer between 15 and -15.
Cutout	$p \in \{0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1\}$, $a \in \{.1, .2, .3\}$, $b \in \{.3, .4, .5\}$	Draw a gray rectangle on the image whose area covers a percentage of the image between a and b.
DLDAE	$p \in \{1\}$, $a \in \{2, 2.5, 3, 3.5, 4, 4.5\}$, $b \in \{5, 5.5, 6, 6.5, 7, 7.5\}$	Encode age labels with α values between a and b. See [48].
Horizontal Flipping	$p \in \{0, .1, .2, .3, .4, .5\}$	Flip the image about the y axis.
Mixup	$p \in \{0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1\}$, $a \in \{.1, .15, .2, .25, .3, .35, .4\}$	Combine two images and their labels. See [56].

fitness of the model that's currently being trained. This forces the algorithm to learn policies that most effectively reduce error specifically for the groups that make up the challenge set. These groups can be seen in table 8. The challenge set contains a total of 5,323 images. 79.8% of the selected images are female. Although the error for males aged 60-79 is relatively high, only $\frac{1}{3}$ of these images are included in the challenge set and they are randomly drawn for each DAP permutation that is tested to help prevent overfitting.

Table 8: The Challenge Set

Subgroup	Count
$\mathbf{G_m} \cap \mathbf{G_f} \cap \mathbf{D_0}$	114
$\mathbf{G_f} \cap (\mathbf{D_1} \cup \mathbf{D_5})$	2345
$\mathbf{G_f} \cap (\mathbf{D_6} \cup \mathbf{D_7})$	625
$\frac{1}{3}\ \mathbf{G_m} \cap (\mathbf{D_6} \cup \mathbf{D_7})$	769
$(\mathbf{G_m} \cap \mathbf{G_f}) \cap (\mathbf{D_8} \cup \mathbf{D_9})$	329
$(\mathbf{G_f} \cap \mathbf{R_b}) - (\mathbf{G_f} \cup \mathbf{D_0} \cup \mathbf{D_1} \cup \mathbf{D_5} \cup \mathbf{D_6} \cup \mathbf{D_7} \cup \mathbf{D_8} \cup \mathbf{D_9})$	1141
Total	5323

Evolving DAPs

The first phase of the evolutionary process involves generating and evaluating a starting population of randomly initialized data augmentation policies. For these policies, between one and three image manipulation techniques are chosen for each subpolicy, and every technique is initialized with settings that are randomly chosen from its reasonable defaults (table 7). Inception-ResNet-v2 is trained for age estimation for 80 epochs on 11,000 images that are randomly chosen from S_{tr}. The reduced training set size allows models to be trained at a rate of roughly 25 per day using four Tesla V100 video cards. Only age estimation is targeted for bias reduction because it has a larger output vector and is a more challenging problem than gender recognition. The policies learned for gender recognition would probably not transfer well to age estimation because they would allow for stronger data augmentation policies that would make it difficult to recognize age. Each epoch, images are loaded and one subpolicy is applied to each image along with the top-level policies. The challenge set is used as vali-

dation data in order to record the lowest loss achieved during training. The validation loss is the fitness score that is used by the evolutionary algorithm to determine which candidate should be mutated. Aging evolution is used as described in [1] which involves recording a population and history of DAPs and their fitness scores. The population has a fixed size of 50. During the evolutionary process, $\frac{1}{3}$ of the population is drawn, and the most fit candidate is chosen to be mutated and evaluated. Three different mutation types are used. The first is a setting mutation where one of the settings for one technique is randomly redrawn from its reasonable defaults. The second is a technique mutation where one technique is chosen to be replaced by a new randomly initialized technique. The third is a size mutation where the number of techniques per subpolicy is increased or decreased. If the subpolicy size is decreased, one of its techniques are simply deleted. If the subpolicy size is increased, a new technique is randomly initialized and appended to the subpolicy. The top-level policies are included only in the settings mutation. The average, minimum, and maximum validation losses of the starting population are 0.0141, 0.0122, and 0.0146 respectively. These numbers for the final population are 0.0092, 0.0086, and 0.0103 which shows a marked decrease in error through evolution. The best discovered policy is illustrated in figure 11. The most commonly chosen data augmentation techniques were random rotation and random Gauss. Random rotation even monopolized an entire subpolicy so, in retrospect, it should probably become a top-level policy. Every data augmentation technique is used at least once which indicates that a wider breadth of techniques is optimal for generalizing models. The bounds on the technique settings are relatively similar for techniques that are chosen more than once. The settings tend to apply mild image manipulations indicating that stronger manipulations make the data unusable. Most subpolicies grew larger in size which rapidly increases the number of potential images that could be generated by a policy.

Figure 12: A random sample of training images with the top 5 data augmentation policies applied.

Results

Final models were trained for age and gender using the top five learned augmentation policies. For the final models, while images are being loaded, one of the top five DAPs is chosen for each image. As shown in [54], using more than five DAPs has little further effect on the performance of the model. Figure 12 provides some examples of what training images look like before they are standardized and then fed into the network. The results in tables 9, 10, and 11 show that age error is reduced in every single subgroup except for D_3 males and females, and females in the wiki dataset. Age estimation error decreased more for R_w than it did for R_b suggesting that it is more challenging to recognize the age of black people. This correlates with the face brightness results which indicate that darker faces are much more challenging to recognize even though there is more data available for them than for the light faces. In fact, although B_1 is by far the most populous face brightness subgroup, results are best for B_2 which is the least populous subgroup. Overall, MAE went from 4.62 to 4.21 with a final MAE of 4.13 for males and 4.90 for females. The overall gender accuracy went from 98.44% to 98.92%. The gender model was additionally evaluated on the PPB dataset

which was held-out during the entire experiment and was not used as training or validation data. Results for PPB are reported in table 13 in the same way that they are reported in [27]. The bias-mitigating effect can be seen to improve gender recognition accuracy by 4.7% for females without a decrease in male accuracy. It improved the results for darker-skinned people much more than it did for lighter-skinned people, and the subgroup that showed the most improvement was dark females. Despite these good results, D_6 and R_b females in our dataset actually showed an increase in error even though they were included in the challenge set. This indicates that a data augmentation policy may be effectively transferred from age to gender to improve accuracy and mitigate bias, however, to achieve the strongest mitigating effects, the evolutionary algorithm would have to be run again to discover policies that work best for gender.

The final MORPH-II MAE was 2.835 and the gender recognition accuracy was 99.60%. These are the best known results for generalized age and gender estimation models that have not been fine-tuned to overfit the dataset [9] [57].

Table 9: Final Results by Decade

	$S_{te} \cap G_m$ Male Age MAE — σ	$S_{te} \cap G_f$ Female Age MAE — σ	$S_{te} \cap G_m$ Male GR Error	$S_{te} \cap G_f$ Female GR Error
$\mathbf{D_0}$	7.33 — 9.69	8.42 — 10.01	17.02%	5.80%
$\mathbf{D_1}$	2.82 — 4.79	5.18 — 7.04	2.71%	2.38%
$\mathbf{D_2}$	3.27 — 4.80	3.77 — 5.23	0.60%	1.37%
$\mathbf{D_3}$	3.79 — 5.10	4.17 — 5.47	0.78%	1.19%
$\mathbf{D_4}$	3.86 — 5.40	5.18 — 6.95	0.51%	1.93%
$\mathbf{D_5}$	5.05 — 6.67	7.18 — 9.51	0.40%	2.86%
$\mathbf{D_6}$	5.75 — 7.84	7.85 — 10.89	0.30%	4.76%
$\mathbf{D_7}$	7.29 — 9.54	9.25 — 11.85	0.00%	5.32%
$\mathbf{D_8}$	11.16 — 13.18	12.03 — 15.46	0.00%	8.70%
$\mathbf{D_9}$	15.66 — 17.01	17.43 — 19.20	3.70%	28.57%

Table 10: Final Results by Race and Illumination

	$S_{te} \cap G_m$ Male Age MAE — σ	$S_{te} \cap G_f$ Female Age MAE — σ	$S_{te} \cap G_m$ Male GR Error	$S_{te} \cap G_f$ Female GR Error
$\mathbf{R_b}$	2.43 — 3.25	3.35 — 4.42	0.31%	1.36%
$\mathbf{R_w}$	2.31 — 3.07	2.82 — 3.78	0.00%	0.91%
$\mathbf{R_o}$	2.43 — 3.26	2.85 — 3.63	0.00%	0.00%
$\mathbf{B_0}$	5.13 — 7.03	5.75 — 7.76	1.36%	3.66%
$\mathbf{B_1}$	3.48 — 5.44	4.57 — 6.33	0.91%	1.52%
$\mathbf{B_2}$	2.94 — 4.30	3.67 — 5.17	0.00%	1.69%

Table 11: Final Results by Dataset

	$S_{te} \cap G_m$ Male Age MAE — σ	$S_{te} \cap G_f$ Female Age MAE — σ	$S_{te} \cap G_m$ Male GR Error	$S_{te} \cap G_f$ Female GR Error
M	2.40 — 3.22	3.18 — 4.22	0.27%	1.10%
I	4.62 — 6.38	4.75 — 6.58	1.03%	1.31%
W	4.92 — 6.74	5.50 — 7.46	0.80%	5.25%

Table 12: Final Gender Scores

	$S_{te} \cap G_m$ Male	$S_{te} \cap G_f$ Female
Precision	99.26%	98.25%
Recall	99.12%	98.52%
$\mathbf{F_1}$	99.19%	98.38%

Table 13: PPB Gender Accuracy

	All	F	M	Darker	Lighter	DF	DM	LF	LM
Baseline	95.8%	90.8%	99.8%	93.9%	97.1%	87.3%	99.6%	93.3%	100%
Final	97.9%	95.5%	99.8%	96.4%	98.9%	92.6%	99.6%	97.5%	100%

Conclusions

The results of this work show that bias in an AI model can be reduced without sacrificing model performance as a whole. In fact, performance increased for almost every single subgroup. State-of-the-art results are obtained for age and gender on the MORPH-II dataset, and results for the IMDB and Wiki datasets are reported for the first time in order to provide an idea of general performance. Running the evolutionary algorithm and evaluating hundreds of models is a computationally expensive process, but further tuning of the DAPs suggested in this paper could result in even better bias-mitigating results. This work shows that DAPs can be transferred from one task to another on the same dataset to reduce bias with some success, however, a deeper exploration into this effect is warranted to understand the dynamics at play. Algorithmic bias should be a standard metric for evaluating data-driven models. It is not enough to report overall performance measures because, as has been proven in the popular press, these metrics do not identify algorithmic bias.

EVOLUTIONARY OVERSAMPLING

This experiment involves the use of dataset oversampling to mitigate bias in a Body mass index (BMI) estimator. BMI is a metric that has been commonly used to identify health risks and to determine a target amount of weight that an individual should lose or gain to be considered healthy. The WHO (world health organization) reports that 2.8 million people die each year because of an overweight or obese BMI. An estimated 35.8 million DALYs (disability-adjusted life years) are lost each year due to overweight [58]. Overweight and obesity have been associated with ailments such as coronary artery disease, diabetes, gallbladder disease, hypertension, sleep apnea, and many types of cancer [59]. In addition to overweight, underweight also possesses associated risks. Underweight individuals may suffer acute energy deprivation or bone density loss, as well as requiring more hours of nighttime sleep [60]. Physical attractiveness has been linked to BMI with the healthiest people being the most generally attractive [61]. Kocabey et. all examine BMI's correlation to social media popularity and find that underweight and obese men have the fewest followers, while underweight women have the most followers [62]. Some countries have legislation which mandates that fashion models be at least above a certain BMI to prevent the spread of eating disorders. BMI is a function of height and weight and can be calculated using the formula:

$$\text{BMI} = \frac{w}{h^2} \times 703 \tag{11}$$

where w is weight in pounds, and h is height in inches. The coefficient 703 converts BMI from an imperial measurement to metric, thus BMI has the unit suffix of $\frac{kg}{m^2}$. There have been some criticisms of using BMI as a metric because it is less accurate as a health indicator on very short or very tall people and individuals with a very low or very high body fat percentage. For example, a tall person might have a higher BMI due to weight but might actually be at a lower health risk than a short person with the same BMI. Because of this,

alternative variations of BMI have been proposed [63]. Classification of BMI varies between organizations, mostly dependent upon the demographics of the population they represent or the treatment objectives of the organization; however, the most commonly recognized BMI ranges are as seen in table 14.

Table 14: BMI Categories

Category	**Range**
Underweight	BMI < 18.5
Normal	$18.5 \leq$ BMI < 25
Overweight	$25 \leq$ BMI < 30
Obese	$30 \leq$ BMI

It has been widely accepted that BMI can be inferred from visual facial cues [64] [65]. Recently, there has been a growing need to estimate BMI from just a photograph. A retail chain might want to gather demographic information on customers including how BMI affects buying habits. A social media company might want to correlate BMI with interactions between users. A life insurance company might want to provide a quote for an interested customer without the lengthy and intrusive process of running blood tests. In addition, using using AI to diagnose patients and recommend personalized treatments is becoming more common. All of these situations require accurate visual verification of BMI, particularly when the estimate needs to be performed remotely without knowledge of the subject's true weight and height. Like other machine learning systems, BMI estimators are prone to algorithmic bias. Female weight tends to vary more than male weight, and male height tends to vary more than female height, but BMI estimators typically perform far worse on women [34]. They also tend to perform better on white people than black people, and better on people with normal BMIs than people with underweight or obese BMIs. Such bias could lead to higher costs for products or services for people of certain subpopulations, or even a larger number of health misdiagnoses. This study uses the large MORPH-IV dataset to examine bias in a BMI estimator. Dataset stats can be seen in figure 13 and tables 15 and 16 below. It should be noted that the average BMIs are much lower than the WHO averages reported

for Americans. We believe this is because of the age demographics of arrestees, and the fact that they usually fall into a lower income bracket. The size of each subgroup can be seen in table 15. This is important because these are the same categories for which weights will determine how often to sample a member of one of these subgroups. We propose an evolutionary strategy for oversampling certain groups from the dataset as a solution for mitigating bias.

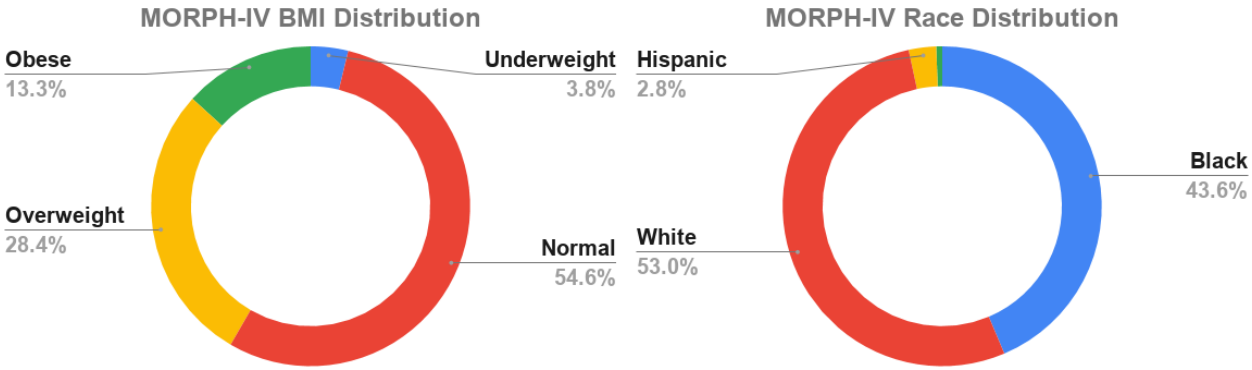

Figure 13: The race and BMI distributions of MORPH-IV. The green sliver in the race chart represents all other races including Asian, Indian, Unknown, and Other.

Table 15: Subgroup Sizes

		Underweight	Normal	Overweight	Obese
Black	**Female**	1340	10495	6072	5320
	Male	1769	74221	47325	22224
White	**Female**	9715	64022	21654	13094
	Male	1565	55764	30282	8851
Hispanic	**Female**	72	1020	453	335
	Male	113	4131	3357	1265
Other	**Female**	52	447	137	51
	Male	55	908	506	178

Baseline Results

Baseline models for BMI estimation were trained using the NASNet-A (6 @ 4032) architecture. NASNet was one of the earliest and most successful attempts at evolving a

Table 16: Average Weights and Heights (Lbs. and In.)

	Weight		Height		BMI	
	Female	Male	Female	Male	Female	Male
Black	156.51	178.21	64.82	69.80	26.19	25.71
White	142.74	171.27	64.74	69.83	23.94	24.69
Hispanic	145.00	165.70	63.28	67.09	25.46	25.89
Other	132.48	158.97	63.04	67.2	23.44	24.75

network architecture that had transferable state-of-the-art performance on CIFAR-10, ImageNet 2012, and COCO [66]. The authors use a recurrent neural network (RNN) to sample architectures from the NASNet search space, which are then trained to a specified accuracy and the RNN is updated accordingly. NASNet blocks are layered and scaled up to a size indicated by the trailing notation. For example, NASNet-A (6 @ 4032) has 4032 leading 3x3 2-stride convolutional filters followed by 6 reduction and normal cells (three each). The first reduction block is doubled presumably so that the tensors don't grow too large when using large batch sizes. We replace the top layers with a global average pooling layer followed by a single ReLU regression node to yield BMI predictions. The dataset is divided into three parts: 80% for the training set, 10% for the validation set, and 10% as a held-out test set. There is no overlap between subjects in the dataset splits. The adadelta optimizer is used alongside mean squared error (MSE) loss. MSE loss was chosen in order to incur a higher penalty for the most erroneous predictions. In addition to just feeding in raw images, the data augmentation policies (DAPs) from [10] are applied. Tables 17 - 19 show the baseline results both with and without data augmentation. Bias is examined by gender, race, and weight class, but not age, because age did not appear to have an impact on BMI bias even in sorely underrepresented subgroups. As can be seen, baseline predictions are generally better for men than women, and the best predictions are made for white men while the worst are made for black women. Additionally, model performance on obese people is far worse than the others because of the extensive range that obesity covers and the difficulty in detecting very high BMIs. Image count is clearly not the only factor in model bias. Predictions are

better for Hispanic and other females than for black females even though there are nearly 10× as many black females in the dataset as Hispanic and other females combined. Some of the bias is thought to arise from the self-reported stats. Most people do not keep track of their weight on a daily basis so it could have changed dramatically from the last time they were weighed. In addition, men are likely to overreport their weight and height to seem more formidable to the police, while women might underreport weight out of shame. To determine how biased a model is, a σ score is calculated by simply taking the standard deviation of the MAEs for each race/gender category along with the weight class categories. A model with the same overall accuracy but higher bias will have a higher σ score. The DAP shows a slight improvement in overall accuracy, but a great improvement in bias, mostly due to the drop in obese error.

Table 17: Baseline MAE by Gender

	Female	**Male**	**Overall**	σ
Baseline	2.82	2.39	2.54	0.93
w/ DAP	2.72	2.29	2.45	0.77

Table 18: Baseline MAE by Race

	Baseline		**w/ DAP**	
	Female	**Male**	**Female**	**Male**
Black	3.44	2.43	3.19	2.36
White	2.67	2.33	2.62	2.18
Hispanic	3.10	2.37	2.97	2.43
Other	2.74	2.59	2.64	2.43

Table 19: Baseline MAE by BMI Class

	Underweight	**Normal**	**Overweight**	**Obese**
Baseline	3.68	1.87	2.32	5.38
w/ DAP	3.76	1.90	2.21	4.64

Evolutionary Oversampling

As previously stated, several attempts have been made to combat dataset bias by oversampling certain parts of the dataset. Many binary classification datasets contain an underrepresented group, and data scientists found that sampling and/or augmenting those groups with a higher frequency led to better results. Systems like SMOTE and ADASYN took an evolutionary approach to learning which individual examples should be sampled the most often. This work proposes a novel approach to learned dataset sampling with the intent of reducing model bias. Given the baseline results, there seems to be a strong correlation between error and BMI standard deviation. White Male BMI standard deviation is the lowest at 4.06, and black female BMI standard deviation is the highest at 6.44. As such, rather than naively sampling dataset subgroups at an even rate, we propose optimizing a dataset sampling weighting scheme against the challenge set described in the next paragraph.

The Challenge Set

A set of particularly challenging images was assembled from the validation set by identifying all images for which baseline BMI estimation error was greater than 4. There are 7,119 images in the challenge set, and it is more evenly balanced than MORPH-IV. The female population rises from 33.97% to 43.78% indicating that there are a high number of largely inaccurate estimations for female BMI. The normal and overweight classes shrink in size while the obese and underweight classes become more prominent. Additionally, there are more black people and fewer white people in the challenge set.

Optimizing Weights

The purpose of this work was to reduce bias by gender, race, and weight class, so a weighting scheme was designed to give a sampling rate to each subpopulation of these categories. There are two genders, four races, and four weight classes yielding a total of 32 subgroups such as “overweight black male” or “underweight hispanic female”. The dataset

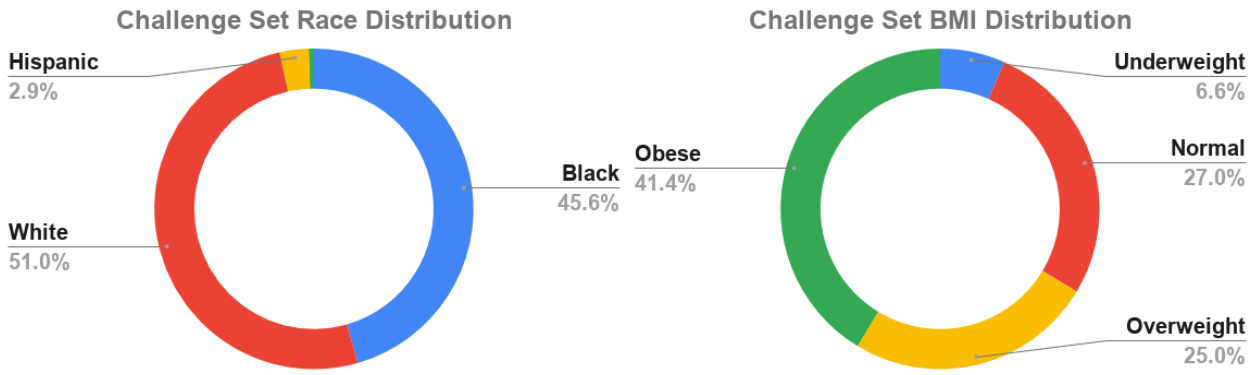

Figure 14: The race and BMI distributions the challenge set. The challenge set is composed of validation set images with the highest error.

is oversampled according to the weights. Every time an image is drawn from the training set, three random values are generated for gender, race, and weight class, and the draw decision is made by the weights. If a specific category is exhausted from the training set, for example, no more "obese other females" are left, every example from that category is added back to the training set and the training set is shuffled. There are 10 weights in every oversampling policy (OSP) – two for gender, and four for both race and weight class. We chose not to give each individual subgroup a weight to minimize the search space. The aging evolution algorithm from [1] is used to optimize weights for sampling the dataset. A starting population of oversampling policies is generated by randomly choosing integers between zero and twenty. If the weight for male is 8, and the weight for female is 12, then there would be a 60% chance that a female will be drawn from the dataset. During evolution, 100,000 images are drawn from the dataset, and only those images are used to train a sample model and evaluate the oversampling policy. Models are trained for 35 epochs, and the best MSE obtained during training is recorded as the fitness of the policy. During evolution, data augmentation is not applied to the images. Due to the greater variation caused in the training set by data augmentation, overfitting is delayed for longer which means that models would have to be trained for many more epochs and the weights would not have as much time to evolve. Additionally, we wanted to see if the learned weights could mitigate bias without the use of data augmentation. This also implies that the data augmentation policies are not

evolved to fit BMI estimation. The evolutionary process consists of keeping track of a population which consists of the most recent 50 evaluated policies and their fitness score which is simply the MSE of the model on the challenge set. From the population of 50 policies, a subset of 16 policies are drawn, and the fittest candidate is mutated by adding 1 to a weight and training a model using the mutated candidate on the dataset subset. Over time, adding 1 to a weight has less of an impact on the sampling rates because the weights grow higher and higher. This process is evolutionary by nature, but should not be considered a genetic algorithm since it does not include candidate crossover. A total of 617 oversampling policies were evaluated during the random sampling and evolutionary process. The fittest candidate can be seen in figure 15.

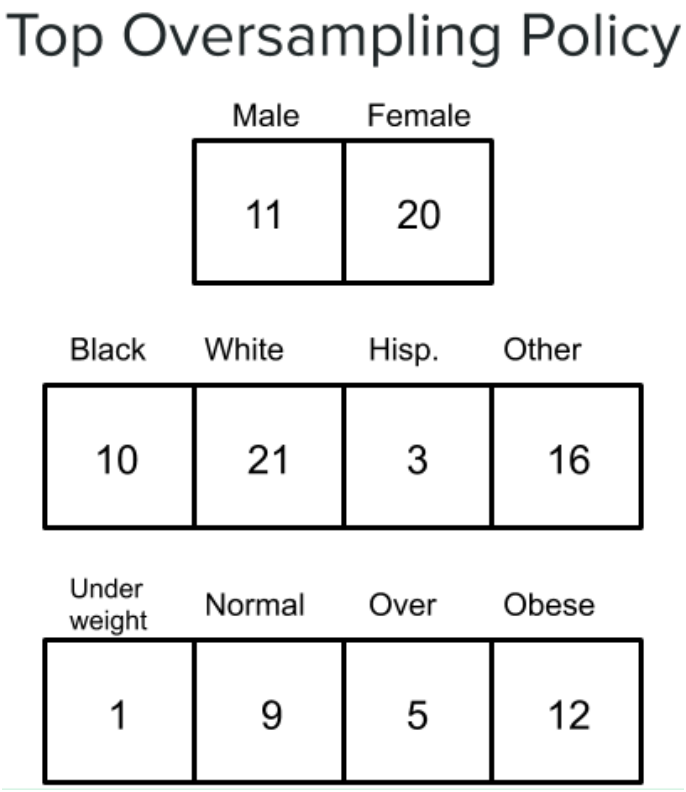

Figure 15: The fittest policy discovered during evolution. In this example, obese black females have the highest chance of being drawn at 12.04%. Interestingly, the female oversampling rate is close to double that of males, and there are also nearly double the number of males in the dataset.

Final Policies

Final models were trained using the same NASNet configuration as the baseline with the top OSP applied. A different model is trained using the top OSP with the addition of the same DAP used earlier. During training, 60,000 images are drawn at a time using the sampling rates determined by the OSP weights. For the DAP model, data augmentation is applied to all the images. The top OSP has a male:female ratio of 11:20, so roughly 65% of the images drawn will be female. With the high "other" sampling rate, some individuals will appear several times in the same training epoch but the images might vary slightly due to the application of the DAP. Final models are trained until training loss grows very close to zero and validation error ceases to improve. The top 5 models by validation loss are saved during training. These models are evaluated on the test set and their σ value is recorded. Model bias tends to shift around during training, so even though the overall results for each of the top 5 models only differs by an MAE of 0.05, σ can change dramatically. For the top 5 models yielded by training with the OSP but not the DAP, σ ranged from 0.96 to 1.02. Training with just the OSP caused overall accuracy to drop slightly, from 2.54 to 2.73, while σ went from 0.92 to 0.96 indicating that the model became more biased overall. Interestingly, however, training with the OSP yielded the best challenge set MAE as seen in table 23. This effect likely means that the OSP model was strongly overfit to the challenge set. The baseline model without data augmentation performed the worst on the challenge set with an MAE of 6.43 while the top unbiased model yielded by training with the top OSP plus data augmentation had a challenge set score of 4.83. Accuracy for women improved more than it did for men. Male accuracy improved by 5.4% while female accuracy improved by 6.0%. The race and gender categories that showed the most improvement were black women and other men which had error decreases of 7.1% and 10.4% respectively. Hispanic females also rose by 6.2% but Hispanic male accuracy dropped by 1.9% and was the only category that did not show improvement. The normal BMI category showed no improvement, the overweight category showed only slight improvement, but the obese category showed

dramatic improvement. Notably, the overall performance of the model only drops by an average error of 0.14 from 2.54 to 2.40, but the final model with data augmentation can be seen to be much less biased.

Figure 16: Example training images with the top OSP and DAP applied.

Table 20: Final MAE by Gender

	Female	**Male**	**Overall**	σ
Top OSP	3.04	2.60	2.76	0.96
w/ DAP	2.65	2.26	2.40	0.72

Table 21: Final MAE by Race

	Top OSP		**w/ DAP**	
	Female	**Male**	**Female**	**Male**
Black	3.55	2.68	3.20	2.34
White	2.94	2.49	2.54	2.13
Hispanic	3.21	2.37	2.91	2.42
Other	2.52	2.59	2.65	2.32

Reflection

Earlier in this study, we trained a midpoint model on the top discovered policy at the time and found that it actually caused an increase in accuracy while reducing bias as seen in figure 17. Needless to say, the authors of this paper were chagrined to find that further

Table 22: Final MAE by BMI Class

	Underweight	**Normal**	**Overweight**	**Obese**
Top OSP	3.91	2.06	2.52	5.65
w/ DAP	3.63	1.87	2.22	4.44

Table 23: Challenge Set Error

Baseline	**Top OSP**	**Top OSP + DAP**
6.43	4.00	4.83

evolution of the oversampling policies caused an overall performance hit. We think it would be remiss to not explain this effect. The authors chose a large sample size to be drawn during the evolutionary process so that even the smallest categories, such as underweight "others", which only have a total of 107 examples, would contain at least a few dozen examples. This slowed down the evolutionary process so we chose a low number of epochs (35) for which to train the sample policies. Because of the high image count and low epoch count, models were not being exhaustively trained, so the evolutionary algorithm was actually learning which policies reduced bias the fastest rather than reducing bias the most overall. Also, as seen in the previous section, it seems that the evolutionary process learned to overfit the challenge set strongly by drawing the same individuals over and over again. If this experiment were to be repeated or applied to a different task, a smaller image count such as 40,000 images would be used for sample models along with doubling the epoch training time. Additionally, the low number of mutated policies meant that the evolutionary algorithm did not have very long to work out a near-optimal policy. Intuitively, the low sampling rates for underweight and Hispanic people would increase over time, while drawing from the "other" category should decrease since it contains so few examples and the "normal" category weight should also decrease since it is the easiest to predict. Further time to run the evolutionary algorithm should result in much better OSPs. These few tweaks to the experiment should allow OSPs to increase overall accuracy without the use of DAPs while also having a bias-mitigating effect.

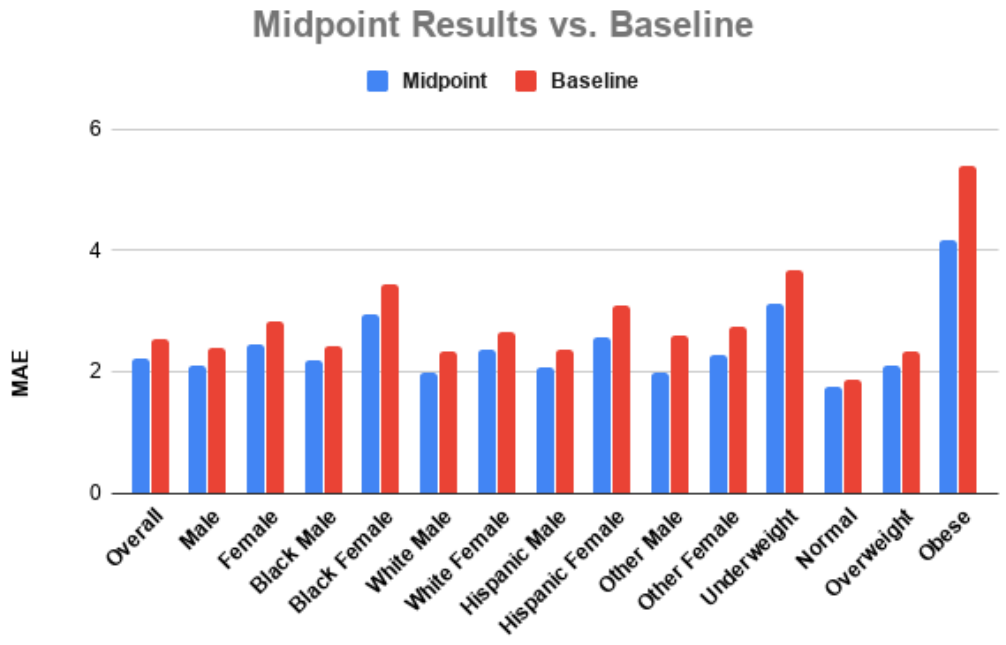

Figure 17: Midpoint results.

NASNet BMI Features

In an effort to demystify the black box, we analyze features extracted by NASNet. For figure 18, we extract features from two sources. The first is the top of reduction block 1. It produces a 60x50x42 tensor which has undergone the first 3x3 2-stride convolutions followed by the reduction block, bringing the feature dimensions down to 50x60. The features are mapped to a monochrome image by subtracting the minimum feature value and multiplying by a scalar equal to $\frac{256}{\max - \min}$ where min and max are the minimum and maximum feature values. The bottom images are the final 4032 7x8 features that occur at the top of NASNet before global average pooling is applied. As can be seen, there is slightly more white in the overweight black male features leading to a higher BMI prediction. The yielded predictions of 18.37 for the white woman and 27.49 for the black man. We also apply occlusion to test set images to determine the facial regions that have the most impact on BMI predictions. First, we look for gender differences. While all regions of the image contribute to the final outcome, the central region of the face from eyes to chin affects predictions the most. Figures 19 and

20 show the percentages of errors caused by occluding different face regions with respect to the min and max errors. For women, the CNN focuses much more on the nose and lip regions than for men. Heatmaps are also generated for white and black test images along with each BMI category. For underweight and normal BMIs, NASNet uses very similar regions to estimate BMI. The overweight and obese categories also share many overlapping regions. Interestingly, the lower BMI classes use opposing regions of the face from the higher BMI classes. Most of the face is used for white females while most of the face is ignored for black females. This is perhaps why the performance is worst for black women. Black and white males share overlap in the most prominent regions but for white males, the lower-right jawbone is given heavier consideration.

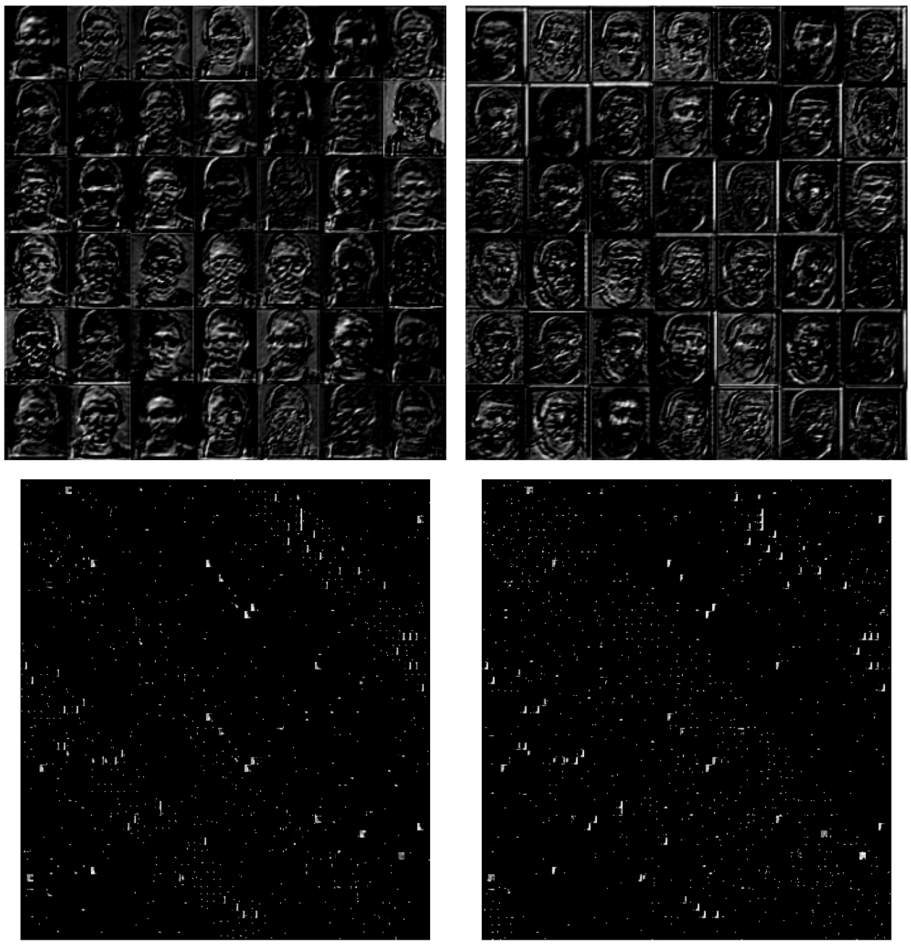

Figure 18: Top: The features yielded by reduction block 1 for a normal white female and an overweight black male. Bottom: The final features yielded at the top of the network for the same subjects.

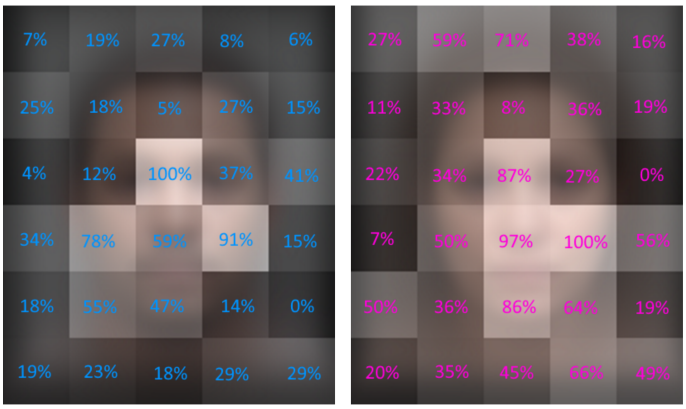

Figure 19: These heatmaps overlaid on male and female mean face images show which regions of the face are the most important in making BMI estimations.

Conclusions

Visual BMI estimation is an increasing important study that has applications in automated demographic analyses. In this study, we trained baseline models for BMI estimation using the MORPH-IV dataset and analyzed the model for bias. We developed a strategy for mitigating bias and reanalyzed the model bias. While some things didn't work out quite the way we wanted them to, we provided a reflection that could aid in designing future experiments. Our final results demonstrate that bias can be reduced by oversampling certain subgroups from the dataset while also reducing overall error. Our top models have the best published accuracy reported for estimating BMI from face images.

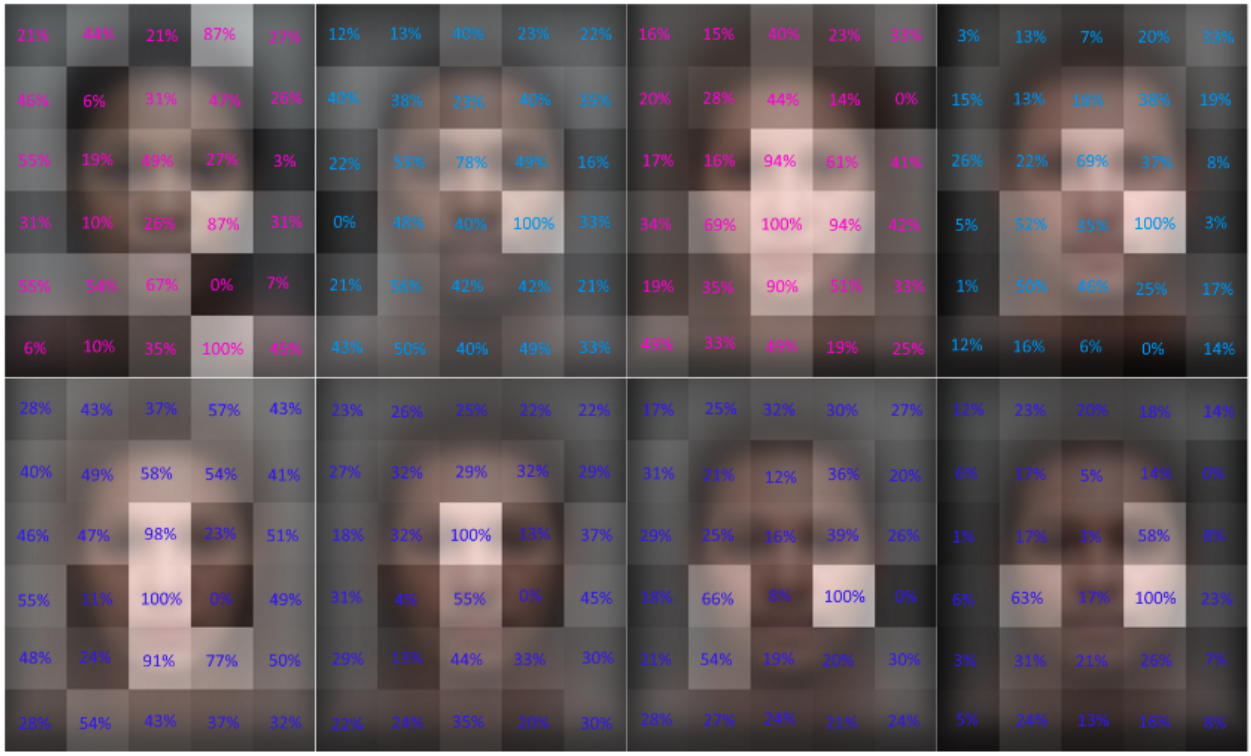

Figure 20: Top-left to bottom-right: Black female, black male, white female, white male, underweight, normal, overweight, and obese heatmaps.

SYNTHETIC DATA

This third and final experiment involves the use of a GAN to generate more data for a dataset to even out the dataset demographics. Specifically, the MORPH-II, IMDB, and Wiki datasets are augmented using the HRFAE GAN and then gender models are trained on the combination of real and generated data. In this experiment, two approaches are taken to balance the dataset: balanced age and balanced gender. Out of the box, HRFAE GAN supports age editing for 1024x1024 images (hence "high resolution") and can edit the age of an individual to appear to be any age between 20 and 70. Because they incorporated Dex into the loss function of the GAN, it probably does not work that well on ages under 20 or over 70 because of the limited child and elderly data that was used to train Dex. First, all of the dataset images are cropped down to 200x200 while keeping the face centered in the image. Then, they are resized to 256x256 using bicubic interpolation. The dataset is randomly shuffled and 5% of it is used as validation while 10% is used as a held-out test set. The number of validation images is kept small to speed up the training process and 5% is still enough to get results that are very statistically significant. For the balanced age training set, images are selected from the more populous age groups and then their ages are edited to fit into the less populous age groups. Any age group between 20 and 70 with less than 2,800 people will be filled out to 2,800 by selecting random examples from groups with more than 2,800 people. The resulting training set has a size of 221,824 images. The original dataset has roughly a 2:1 male:female ratio, so to balance gender, every female example between the ages of 20 and 70 has her age randomly edited and is added to the dataset resulting in a balanced gender dataset of 245,364 images. Now, not only are their more training images, but the training set has been balanced for age and gender. Balanced age and balanced gender models are trained on their distinct training sets while saving the top 5 models. NASNet-A is used again with the same training parameters as in section except that instead of one regression node at the top of the network, there are two softmax nodes – one male and one

female. Data augmentation is also applied using the learned policies in [10]. The biggest difference between the balanced age and balanced gender models is in $\mathbf{D_0}$ females where the balanced age model had a 4.55% error rate and balanced gender had no error. There is also a big shift in $\mathbf{D_6}$ where male error increases slightly but female error halves. The balanced gender model had the best performance with an overall accuracy of 99.82%. Looking back in section , it can be seen that the baseline results on the same dataset were 98.44%, which is a good starting point, but with the application of data augmentation and synthetic data, accuracy is raised by 1.38%. Finally, the MORPH-II test set accuracy hit 100% making these synthetic models the new state-of-the-art.

Table 24: Test Set Gender Error

	Balanced Age		**Balanced Gender**	
	Male	Female	Male	Female
$\mathbf{D_0}$	0.00%	4.55%	0.00%	0.00%
$\mathbf{D_1}$	0.49%	0.95%	0.49%	0.57%
$\mathbf{D_2}$	0.12%	0.28%	0.26%	0.08%
$\mathbf{D_3}$	0.15%	0.37%	0.17%	0.12%
$\mathbf{D_4}$	0.00%	0.17%	0.12%	0.09%
$\mathbf{D_5}$	0.00%	0.80%	0.00%	0.80%
$\mathbf{D_6}$	0.00%	1.01%	0.28%	0.50%
$\mathbf{D_7}$	0.00%	0.00%	0.00%	0.00%
$\mathbf{D_8}$	0.00%	0.00%	0.00%	0.00%
$\mathbf{D_9}$	0.00%	0.00%	0.00%	0.00%
$\mathbf{R_b}$	0.00%	0.00%	0.00%	0.00%
$\mathbf{R_w}$	0.00%	0.00%	0.00%	0.00%
$\mathbf{R_o}$	0.00%	0.00%	0.00%	0.00%
$\mathbf{B_0}$	0.30%	0.88%	0.47%	0.35%
$\mathbf{B_1}$	0.11%	0.35%	0.15%	0.16%
$\mathbf{B_2}$	0.06%	0.00%	0.00%	0.00%
M	0.00%	0.00%	0.00%	0.00%
I	0.25%	0.36%	0.31%	0.14%
W	0.08%	1.11%	0.15%	0.62%
Precision	99.86%	99.60%	99.82%	99.82%
Recall	99.80%	99.72%	99.92%	99.63%
$\mathbf{F_1}$	99.83%	99.66%	99.86%	99.73%

CONCLUSION

Bias in production AIs has recently become highly relevant to a number of industries. While it has been limitedly examined in the past, it has not yet gotten the attention it deserves. As a people, we cannot allow systemic bias such as racism and sexism to continue to be propagated by the computer systems we rely on, and the machine learning models that automate so much of our work. This thesis presented background knowledge related to bias and AI. It proposed that bias can be mitigated even while using biased datasets to learn. It examined evolutionary algorithms to pick an effective strategy for evolving very time-consuming mitigation solutions. Convolutional neural networks were discussed in order to shine light on their inner workings. The datasets used for the experiments in this thesis were fully disclosed, and three experiments were performed which proved to effectively reduce bias without sacrificing quality. In fact, an age MAE of 2.82 is reported for MORPH-II, along with a gender recognition accuracy of 99.82%, both of which are state-of-the-art. Additionally, a BMI MAE of 2.40 is reported, which is also state-of-the-art, however, few visual BMI studies have been performed prior to this one, and they also used far fewer images. While these results could be deemed impressive, AI bias is not a solved problem. The topic demands much more extensive work with larger datasets, a wider variety of tasks, and long-term deployed systems. If we take these steps, we can help ensure fair systems for future generations.

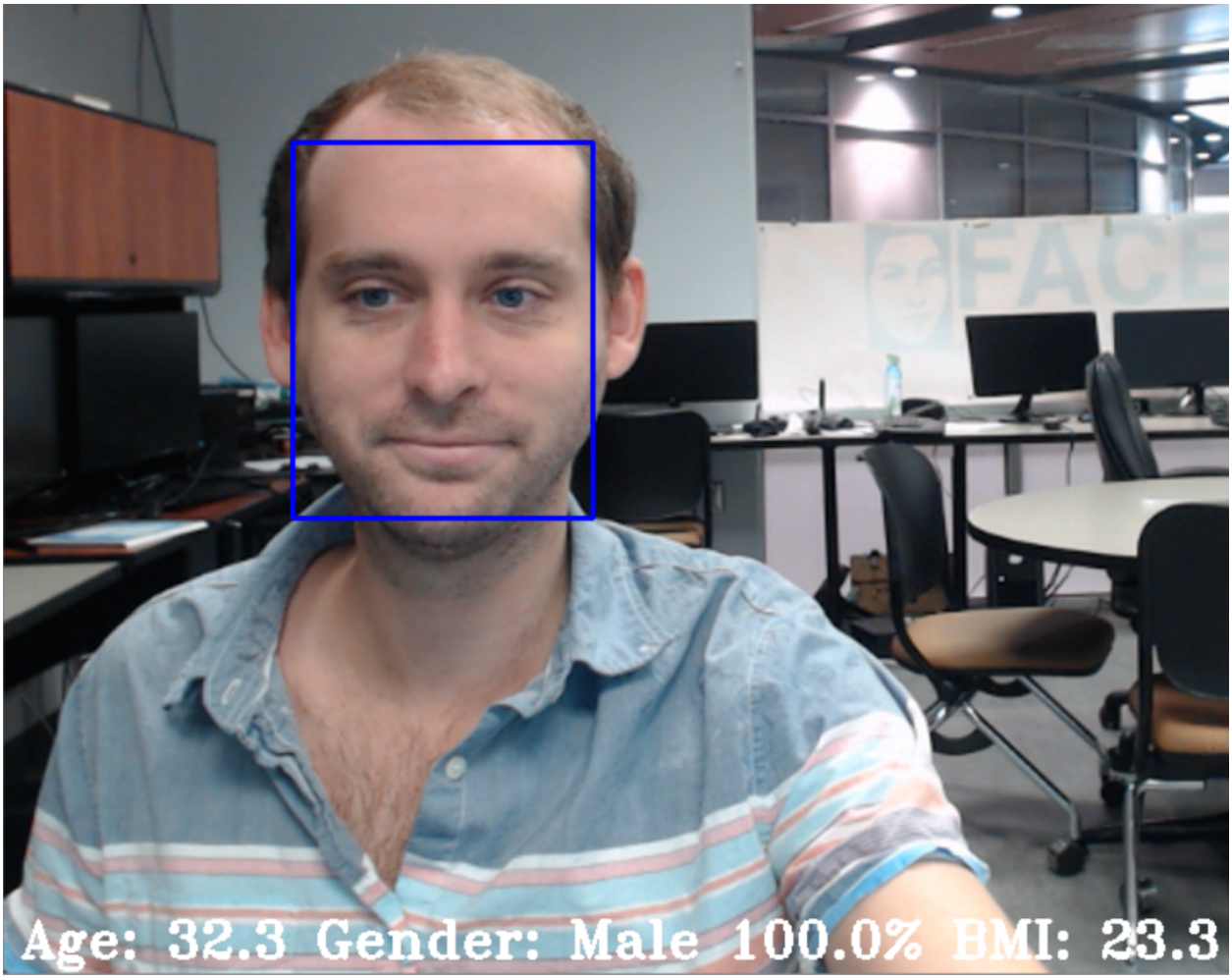

Figure 21: Real-world demonstration of an age, gender, and BMI estimator running on a live webcam feed.

REFERENCES

[1] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," CoRR, vol. abs/1802.01548, 2018.

[2] P. Grother, M. Ngan, and K. Hanaoka, "Face Recognition Vendor Test (FRVT) Part 3: Demographic Effects," Dec 2019.

[3] "Algorithmic Bias," Jun 2020.

[4] Y. Fu, G. Guo, and T. S. Huang, "Age synthesis and estimation via faces: A survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 11, pp. 1955–1976, 2010.

[5] F. Rosenblatt, The Perceptron, a Perceiving and Recognizing Automaton Project Para. Report: Cornell Aeronautical Laboratory, Cornell Aeronautical Laboratory, 1957.

[6] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," Neural Computation, vol. 4, no. 1, pp. 1–58, 1992.

[7] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in CVPR 2011, pp. 1521–1528, June 2011.

[8] G. Panis, A. Lanitis, N. Tsapatsoulis, and T. F. Cootes, "Overview of research on facial ageing using the fg-net ageing database," IET Biometrics, vol. 5, no. 2, pp. 37–46, 2016.

[9] G. Antipov, M. Baccouche, S.-A. Berrani, and J.-L. Dugelay, "Effective training of convolutional neural networks for face-based gender and age prediction," Pattern Recognition, vol. 72, pp. 15–26, 2017.

[10] P. Smith and K. Ricanek, "Mitigating algorithmic bias: Evolving an augmentation policy that is non-biasing," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops, March 2020.

[11] K. Ricanek and T. Tesafaye, "Morph: A longitudinal image database of normal adult age-progression," in Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on, pp. 341–345, IEEE, 2006.

[12] H. Han, C. Otto, and A. K. Jain, "Age estimation from face images: Human vs. machine performance," 2013 International Conference on Biometrics (ICB), 2013.

[13] G. Guo and G. Mu, "Human age estimation: What is the influence across race and gender?," in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, pp. 71–78, IEEE, 2010.

[14] G. Guo and G. Mu, "A study of large-scale ethnicity estimation with gender and age variations," in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, pp. 79–86, IEEE, 2010.

[15] G. Guo, Guowang Mu, Y. Fu, and T. S. Huang, "Human age estimation using bio-inspired features," in 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 112–119, 2009.

[16] G. Guo and G. Mu, "Simultaneous dimensionality reduction and human age estimation via kernel partial least squares regression," in CVPR 2011, pp. 657–664, June 2011.

[17] Y. Wang, K. Ricanek, C. Chen, and Y. Chang, "Gender classification from infants to seniors," in 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), pp. 1–6, IEEE, 2010.

[18] K. Ricanek, Y. Wang, C. Chen, and S. J. Simmons, "Generalized multi-ethnic face age-estimation," in 2009 IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems, pp. 1–6, 2009.

[19] Shuicheng Yan, Xi Zhou, Ming Liu, M. Hasegawa-Johnson, and T. S. Huang, "Regression from patch-kernel," in 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, 2008.

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015.

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

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

[23] R. Rothe, R. Timofte, and L. V. Gool, "Dex: Deep expectation of apparent age from a single image," in IEEE International Conference on Computer Vision Workshops (ICCVW), December 2015.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," CoRR, vol. abs/1512.03385, 2015.

[25] M. Bruveris, J. Gietema, P. Mortazavian, and M. Mahadevan, "Reducing geographic performance differentials for face recognition," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops, March 2020.

[26] T. Bolukbasi, K. Chang, J. Y. Zou, V. Saligrama, and A. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," CoRR, vol. abs/1607.06520, 2016.

[27] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in Conference on fairness, accountability and transparency, pp. 77–91, 2018.

[28] 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, pp. 2672–2680, 2014.

[29] J. Gauthier, "Conditional generative adversarial nets for convolutional face generation," Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester, vol. 2014, no. 5, p. 2, 2014.

[30] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, "Attgan: Facial attribute editing by only changing what you want," IEEE Transactions on Image Processing, vol. 28, no. 11, pp. 5464–5478, 2019.

[31] G. Antipov, M. Baccouche, and J.-L. Dugelay, "Face aging with conditional generative adversarial networks," in 2017 IEEE international conference on image processing (ICIP), pp. 2089–2093, IEEE, 2017.

[32] X. Yao, G. Puy, A. Newson, Y. Gousseau, and P. Hellier, "High resolution face age editing," arXiv preprint arXiv:2005.04410, 2020.

[33] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4401–4410, 2019.

[34] L. Wen and G. Guo, "A computational approach to body mass index prediction from face images," Image and Vision Computing, vol. 31, no. 5, pp. 392–400, 2013.

[35] E. Kocabey, M. Camurcu, F. Ofli, Y. Aytar, J. Marin, A. Torralba, and I. Weber, "Face-to-BMI: Using computer vision to infer body mass index on social media," arXiv preprint arXiv:1703.03156, 2017.

[36] J. Deng, J. Guo, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," CoRR, vol. abs/1801.07698, 2018.

[37] M. Jiang, G. Guo, and G. Mu, "Visual BMI estimation from face images using a label distribution based method," Computer Vision and Image Understanding, p. 102985, 2020.

[38] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," Journal of artificial intelligence research, vol. 16, pp. 321–357, 2002.

[39] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), pp. 1322–1328, IEEE, 2008.

[40] V. Karia, W. Zhang, A. Naeim, and R. Ramezani, "Gensample: A genetic algorithm for oversampling in imbalanced datasets," arXiv preprint arXiv:1910.10806, 2019.

[41] M. T. Jones, Artificial intelligence: a systems approach. Jones and Bartlett Publishers, 2009.

[42] J. Krarup and P. M. Pruzan, "Computer-aided layout design," in Mathematical programming in use, pp. 75–94, Springer, 1978.

[43] P. Hahn and J. Krarup, "A Hospital Facility Layout Problem Finally Solved," Journal of Intelligent Manufacturing, vol. 12, 05 2000.

[44] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," Neural computation, vol. 1, no. 4, pp. 541–551, 1989.

[45] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," IEEE Transactions on pattern analysis and machine intelligence, vol. 20, no. 1, pp. 23–38, 1998.

[46] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," IEEE transactions on neural networks, vol. 8, no. 1, pp. 98–113, 1997.

[47] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "Wider face: A face detection benchmark," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5525–5533, 2016.

[48] P. Smith and C. Chen, "Transfer learning with deep cnns for gender recognition and age estimation," in 2018 IEEE International Conference on Big Data (Big Data), pp. 2564–2571, IEEE, 2018.

[49] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," CoRR, vol. abs/1409.4842, 2014.

[50] L. Taylor and G. Nitschke, "Improving deep learning using generic data augmentation," CoRR, vol. abs/1708.06020, 2017.

[51] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," CoRR, vol. abs/1311.2901, 2013.

[52] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," CoRR, vol. abs/1710.09412, 2017.

[53] T. Devries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," CoRR, vol. abs/1708.04552, 2017.

[54] E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," CoRR, vol. abs/1805.09501, 2018.

[55] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," CoRR, vol. abs/1212.5701, 2012.

[56] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," CoRR, vol. abs/1710.09412, 2017.

[57] R. Rothe, R. Timofte, and L. Van Gool, "Deep expectation of real and apparent age from a single image without facial landmarks," International Journal of Computer Vision, vol. 126, no. 2-4, pp. 144–157, 2018.

[58] "Body mass index (bmi)."

[59] "Managing overweight and obesity in adults," Evidence Report, 2013.

[60] "Physical status: The use and interpretation of anthropometry."

[61] M.-E. Brierley, K. R. Brooks, J. Mond, R. J. Stevenson, and I. D. Stephen, "The body and the beautiful: health, attractiveness and body composition in men's and women's bodies," PLoS One, vol. 11, no. 6, 2016.

[62] E. Kocabey, F. Ofli, J. Marin, A. Torralba, and I. Weber, "Using computer vision to study the effects of bmi on online popularity and weight-based homophily," in International Conference on Social Informatics, pp. 129–138, Springer, 2018.

[63] N. Trefethen, "New bmi (new body mass index)."

[64] V. Coetzee, J. Chen, D. I. Perrett, and I. D. Stephen, "Deciphering faces: Quantifiable visual cues to weight," Perception, vol. 39, no. 1, pp. 51–61, 2010.

[65] D. D. Pham, J.-H. Do, B. Ku, H. J. Lee, H. Kim, and J. Y. Kim, "Body mass index and facial cues in sasang typology for young and elderly persons," Evidence-Based Complementary and Alternative Medicine, vol. 2011, 2011.

[66] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," CoRR, vol. abs/1707.07012, 2017.