An Approach for Reducing Racial Bias in Facial Monitoring Systems

by

Viktor Ciroski

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Masters of Science in Engineering
(Robotics Engineering)
in the University of Michigan-Dearborn
2023

Master's Committee:

Lecturer Azeem Hafeez, Chair
Professor Selim Awad
Associate Professor Xuan Zhou

ORCID iD 0000-0001-8855-6158

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Azeem Hafeez, for taking me on as a student and helping guide me through the thesis process. I believe tough his mentorship I have become a more independent researcher, and a better writer, and all around has prepared me for my Ph.D. I would also like to thank my employer General Motors for providing the financial support to fund my Master's degree. This commitment made all the difference in my future. Finally, I would like to thank my friends and family for their emotional support throughout this process. I would also like to take this time to apologize and thank you for your patients as I ramble on about my passion for artificial intelligence. This isn't the last time I'll thank you, and I'm grateful every day for your presence.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF ABBREVIATIONS

CNN Convectional Neural Networks

DNN Deep Neural Networks

FMS Facial Monitoring Systems

GPU Graphics Processing Unit

mAP Mean Average Precision

RAM Random-Access Memory

RPN Region Proposal Network

S4L Semi-Supervised Self-Supervised Learning

ABSTRACT

This thesis provides a new approach to reduce racial bias issues and inaccuracies caused by unbalanced benchmark datasets in facial detection systems. It is well known that these unbalanced benchmark datasets significantly over-represent white individuals. It is also understood that a deep learning model performance is based on the data used for training. With these two conjectures, previous research has shown how inaccuracies across different racial groups can be hidden by tracking a single class's labels, faces, and performance. New balanced benchmark datasets have been developed however they lack the variability seen in transitional benchmark sets. Additionally, manually annotating and retraining these models is both computational and financially expensive. Therefore, this research proposed a financially inexpensive way to reduce racial bias within pre-trained facial monitoring systems using semi-supervised self-supervised learning.

CHAPTER I

Introduction

They are watching you. The modern age brings modern surveillance. From companies like META to government agencies, and autonomous vehicles they are watching you. Thanks to advancements in computing hardware and image recognition technology they now have the ability to detect, locate, and identify who you are. But who they are, they are agencies implementing Facial Monitoring Systems (FMS) to isolate the location of faces within images. Do not fear these technologies that are not intrinsically invasive, companies such as Snapchat or TikTok use FMS to properly place graphics over the user's face with filters. This thesis will not cover the arguments for or against the ethical or legal gray area FMS technology exists within. However, the main objective of this thesis is to educate the reader on the limitations of current FMS systems, and the asymmetries of representation of current benchmark datasets, and suggest a methodology to inexpensively correct these asymmetries within pre-existing facial monitoring systems.

The remainder of this section will be structured as follows. Section 1.1 sheds light on the motivations to provide a solution to reduce racial bias within pre-existing FMS. Section 1.2 will provide the justification and outline the objectives of this research followed by the contributions made in section 1.3. This portion of the thesis will then be concluded with a detailed outline for the remainder of this paper.

1.1 Motivation

The industry has been releasing applications to detect human faces for almost over a decade. These models claim to be over 90% accurate in their detection abilities; however, when released to the public this does not seem to be the case. For instance, Facebook, now META, released its facial detection model DeepFace back in 2014. This model was trained on over four million images and claimed to be more accurate at detecting faces than a human being with an accuracy of 97.35% (1). This claim is misleading, one would assume the above human accuracy would imply the model performance would be able to detect all types of people. Although, this is not the case. The claim above human accuracy and 97.35% accuracy refers to the model's ability to detect faces more accurately than humans within the dataset used for training. To further clarify this means that humans may not have been able to see past occlusions in the dataset, or simply that the model's training set was biased. It wasn't publicized till 2020 when users started to notice that then Facebook's facial detection and labeling models were auto-labeling black men as primates (2).

Given the benefit of doubt, in 2014, the year DeepFace was released, there were no widely available or well-known datasets to test a model's performance on different racial groups. However, only four years later, in 2018, Dr. Joy Buolamwini published her work titled, "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification". This work showed how to benchmark datasets were over-representing light-skinned white men over other groups. Dr. Buolamwini went to, "composed of 1270 unique individuals that are more phenotypically balanced on the basis of skin type than existing benchmarks" and further showed how this dataset could be used to benchmark a facial detection models performance across multiple skin types (3). This incident with Facebook is not isolated other social media apps have come under fire as well: Snapchat not recognizing black users (4), TikTok algorithm appears to use the "physical characteristics of a person's profile picture were mimicked in TikTok's recommendations" (5). Therefore, it's safe to say even with new uniformly distributed benchmark datasets publicly available, the industry is either not testing or not publishing the true granular performance of their models.

It's important to note that, while these new benchmark datasets are uniform they do not provide the variability in occlusion, scale, or position that previous face datasets did. Therefore, while these models are ideal for understanding the bias within a system they would be unable to create a robust model needed for industry use. Only a year after Facebook acknowledged the racial bias in their face detection software then vice president of Artificial Intelligence Jerome Pesenti announced the company would shut down the face recognition system (6). Shutting down or halting the development of the face recognition system is not the solution. In order to create a safer and more equitable world, new approaches must be developed that allow for models to be developed in inexpensive and optimal ways.

1.2 Research Objectives

There exist a hole within current training techniques used for FMS. This whole exists because of two main issues.

- Financial justification to train new models
- Financial justification to label new data

The reason robust models have not been developed has never been because of a lack of diverse data, or an engineer's ability to train a robust model. The reasoning seems to be financial. In regards to the data collection pipeline, it is first expensive to collect images of people, an additional cost is tacked on when the idea of uniform representation is considered. After the data has been collected it must be labeled, which can not only be considered a financial burden, but also a considerable time commitment. Finally, after the data is collected and labeled it can be checked for uniform distribution, if this condition is not met more data must be collected, labeled, and checked again. Finally, the model can be trained, with no guarantee the data is able to properly remove the bias.

Therefore a new training technique must be explored, one that is inexpensive. The most time-consuming, and therefore expensive a portion of training a new model is the labeling of data. This manual process requires hundreds of hours of tedious work drawing boxes around a region of interest. Secondly, training a model from scratch requires a large dataset of millions of images and thousands of hours of training time. The objective of this research is to understand if it is possible to use a pre-existing face detection model to automatically label data from a uniformly distributed dataset to reduce racial bias. The proposed method for this research will explore the benefits of Semi-Supervised Self-Supervised Learning (S4L) technique on significantly reducing racial bias across 7 different racial groups. After this, the ANOVA statistical significant test will be used to determine the significance of these findings.

1.3 Contributions

The main contribution of this thesis is a new technique for reducing racial bias within facial detection systems

1.4 Thesis Outline

This thesis is organized into seven chapters.

Chapter II will cover the fundamentals of the main approaches used within this thesis. Each section will provide a brief history explanation of the individual approach. Chapter III contains the literature review for this work. In this chapter this thesis will go over the facial monitoring pipeline, and past work. It will also cover a review of semi-supervised self-supervised learning and how these techniques can be expanded. Finally, the literature review will end with a discussion of different biases that exist within a deep learning model. As the Faster R-CNN architecture is the backbone of the model used within this thesis chapter IV will explore this topic. After which a review of the previous dataset, balanced and unbalanced will be discussed in chapter V. Additionally, this chapter will contain information on the datasets used for training within this thesis. Finally, this paper will end with chapter VI providing information on the training setup and results analysis. Followed by chapter VII on the conclusion and future work.

CHAPTER II

Fundamentals

Artificial intelligence and specifically deep neural networks have revolutionized researchers' ability to create generalized solutions. This section will go over the fundamentals of deep learning in an attempt to demystify the black box that is a neural network. This section will also go over the fundamental learning techniques used within this research: Transfer Learning, and Semi-Supervised Self-Supervised Learning. The goal of this chapter is to provide a foundational background to reference through the remainder of this thesis, explore the history of these techniques, and explain the benefits and drawbacks of each approach.

The remainder of this chapter will be organized with a review of deep learning. This section will start from the basics of the perception and work its way up to Convectional Neural Networks (CNN). This chapter will then transition into the topic of Transfer learning, a very powerful training technique commonly used. Finally, this section will end with an explanation of it's history, technique, and use cases for Semi-Supervised Self-Supervised Learning.

2.1 Review of Deep Learning

The human brain may be considered by some to be the most computationally efficient computer in existence. From an engineering point of view, the human brain

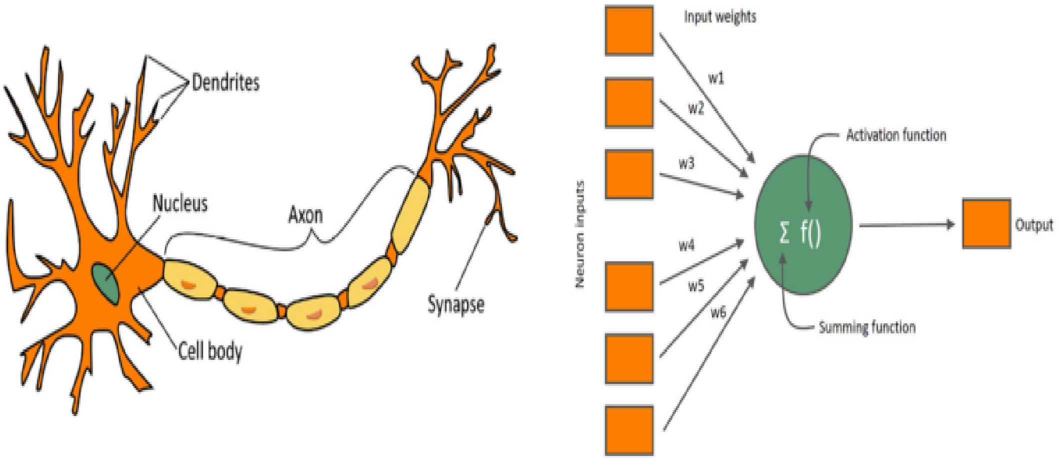


Figure 2.1: Biological and Artificial Neuron Comparison

receives raw input data, from various external sensors, that is then fed through the nervous system to the brain, where the biological neural network can quickly and accurately understand and react to the environment around it. If one could follow a signal through the brain it would look something like the following. The signal enters the brain network through the neuron cell's dendrites. It's fed into the cell body where the same type of input-output mapping function is happening and finally exits the cell through the axon. The question then becomes, if the human brain has a systematic process for input-output mapping, why not a computer? This is exactly what happened, Modern-day artificial neural networks work with the same basic principles as the human brain. Figure 2.1 shows the relationship between a biological neuron and the artificial neuron or the perceptron (7).

Mathematically, deep learning has been possible since the 1980s. However, because of the computational complexity and limitations leading the ”AI Winter” deep learning has only become relevant again in the last decade. Ian Goodfellow et al. state,

> Today, artificial intelligence (AI) is a thriving field with many practical applications and active research topics. We look to intelligent software to automate routine labor, understand speech or images, make diagnoses in

medicine and support basic scientific research (8)

However, even with the resurgence in AI interest, advanced computing, and access to funding deep learning still has many topics to be researched.

2.1.1 The Perceptron

The perceptron is a mathematical linear classifying function. In simple terminology, the perceptron attempts to find a straight line that can map the input data to the output. This is done by taking an input vector of numerical values, performing and an aggregation of the weighted sum in an attempt to find the best combination of weights to perform this mapping. It is understandable to confuse the perceptron with the more well-known sigmoid neuron. The perceptron is a generalized version of the sigmoid neuron. It was first invented by McCulloch and Pitts in their 1943 research at the Cornell Aeronautical Laboratory (9). The early perceptron was only able to solve linearly separable problems, the mathematical formulation of the perceptron can be seen in equation (2.1).

$$y = b + \frac{1 \text{ if } \sum w_i * x_i \geq 0}{0 \text{ if } \sum w_i * x_i < 0} \tag{2.1}$$

where b is the bias, w is the weights, and x is the input vector

2.1.1.1 Weights, Bias and Activation Functions

Equation 2.1 refers to two parameters the bias unit and the weights vector. The magic of the perceptron comes from its ability to learn these parameters automatically. Only by learning the correct combinations of weights and biases, the perception can create is mapping to the output classification. These two parameters are more than just random noise, with the hope that they find the output. The weights can be thought of as expressing how much influence the input values will have on the mapping. For example, a weight approaching zero would indicate there is little to no effect on the output, while a weight of one should be considered to have a substantial influence. The weight parameter specifically is what is updated to find the ideal mapping. The bias unit plays a special role in ensuring that when all inputs are one, the activation function is still operational. This unit has no inputs and is considered static.

$$y = g(b + \sum w_i * x_i) \tag{2.2}$$

where g is activation function

The activation function is what allows a cascading array of artificial neurons to learn more complex patterns. An activation function takes the outputs of previous neurons and transforms them into the next layer's inputs, formally defined in equation (2.2). In general, this transformation will keep the values within a manageable range. Traditionally one of three activation functions are chosen when training a neural network: sigmoid, tanh, and ReLU.

2.1.2 Hyperparameters

A model's ability to learn is not only affected by the quality or quantity of data it is fed. Hyperparameters are a set of attributes used to control the learning process of the model. There is a vast array of learning parameters to consider, but in general the three most common seem to be the learning rate, learning momentum, and epochs.

The learning rate is a positive scalar determining the step size for the learning progress (8). The learning rate is used to optimize the model's capacity, or in other words, the learning rate is used to find a local minimum in the model's ability to understand pattern recognition. The learning rate is traditionally low, within the magnitude of 1e-3 to 1e-5; however, each model requires a different learning rate. When a learning rate is too low, convergence can be slow. On the contrary, when the learning rate is too high, the learning process is accelerated by updating weights with

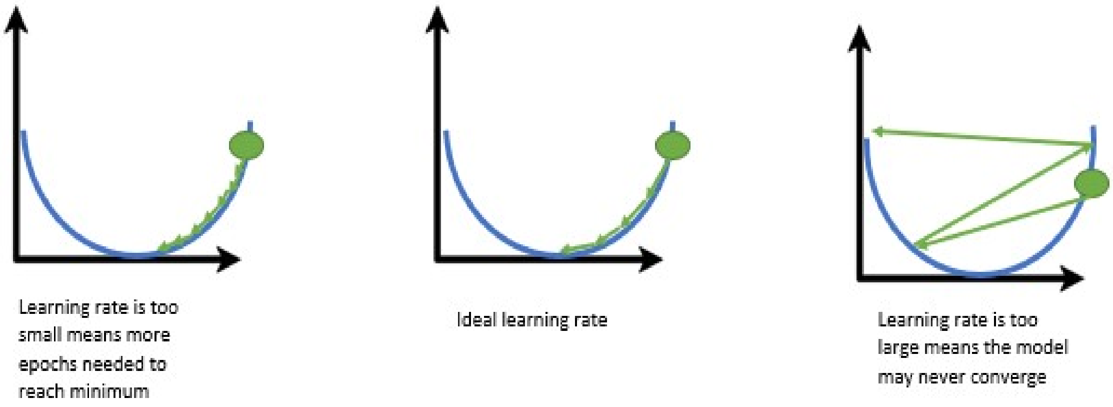

Learning rate is too small means more epochs needed to reach minimum
Ideal learning rate
Learning rate is too large means the model may never converge

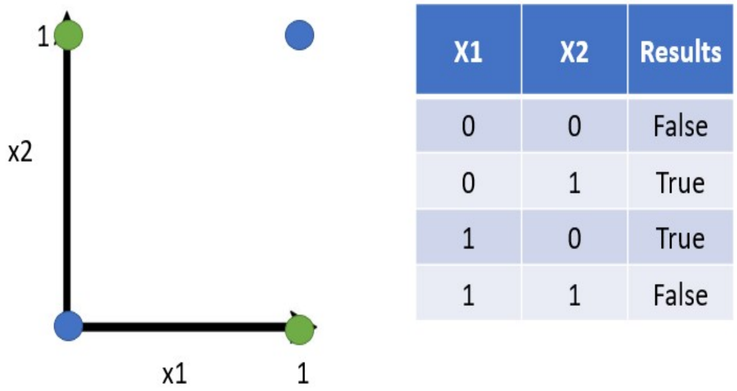
Figure 2.2: Learning Rate

larger values, and therefore it may never converge. These two examples can be seen in Figure 2.2.

There are instances where a model's training curve shows positive performance on the training data, but not on the testing data, such as overfitting. Overfitting happens when a model learns the noise within the training data and cannot generalize to new data. A solution to avoid overfitting is to use regularization techniques. The use of regularization penalizes the loss function while encouraging the learning algorithm to keep the weights an appropriate size. The weights within the network are updated using a learning algorithm, such as backpropagation.

The learning momentum is used to reduce error for the predicted and ground truth targets by converging at a global minimum. Figure 2.2 above shows the ideal error surface; however, the error surface usually contains many local minima, shown in Figure 2.2. Learning momentum helps the learning algorithm to avoid getting stuck within one of the many local minima. This is done by allowing the learning momentum to divide the gradient vector's direction by preventing oscillations while updating the weight values.

The epoch parameter is an integer value used to specify the number of passes the network will go through the whole dataset during training. The training data is passed through multiple times until the error from the model is sufficiently minimized.



X1	X2	Results
0	0	False
0	1	True
1	0	True
1	1	False

Figure 2.3: XOR Function

2.1.3 Deep Neural Networks

As it has been stated, the perceptron alone is a linear classifier. This means that a single perceptron can solve AND OR logic gates with ease. However, the perceptron fundamentally is unable to solve something as simple as the XOR (exclusive or) logic gate. There is no possible way to draw a straight line through the XOR function such that it evenly separates the true and false regions. Figure 2.3 provides a visual of the XOR function. However, Minsky and Paper were able to solve the nonlinear separable problem of the perceptron (10).

The cascading of perceptron nodes brought rise to multilayered neural networks or Deep Neural Networks (DNN). Allowing the perceptrons to be arranged in a cascading array, it was found that the nodes can work together and discover nonlinear functions. This allows the network to revile more complex patterns and classification functions than ever before. The first and last layers are referred to as the input and output layers respectively. All other layers are called hidden. The hidden layers use activation functions to transform the input data into a form the output layer can use. Geometrically one can think of the hidden layers as creating a hyper-plan in higher dimensional space to create separation between data points. Similar to how simple logistic regression finds the best fit line.

Building off the perceptron, the goal of most neural networks is to optimize the weight values to develop the mapping function of the model. There are two ways this can be done. A human could manually set each weight value, check the output, and attempt to find the best combination by force, or they could algorithmically allow the model to do this work itself. For the model to understand how this selection of weights affects its performance there must be a way to understand the expected versus predicted results. This can be done using forward propagation. Forward propagation allows the model to infer how well it is learning, but this alone is unable to create a new set of weight values. To update the weight the model must implement what is called backpropagation. Given the error between the predicted and expected results, backpropagation will calculate the gradient of the model's error function with respect to the model's weights. This gradient is then used to update the model's weight in hopes of reaching a local minimum for learning.

2.1.4 The Convolutional Neural Network

There are many types of deep neural network architectures, each with its name, and generally an associated use case. The CNN, convolutional neural net, is the architecture that is generally associated with use cases involving grid-like data. Such as video or image data. As one might have guessed, the name convolutional neural network comes from the convolution function. This architecture is considered ideal for grid-like data because of the convolution functions' ability to reduce the number of parameters without a loss of quality. The parameter reduction caused by the convolution operation is the bread and butter of CNNs. They are not only faster and more efficient than training a typical DNN, but have been shown to produce better results. CNNs have been hugely successful in practical applications. Because of CNN's history of success and ability to reduce data complexity, they have become a highly used architecture in deep learning. Ian Goodfellow et al. state that

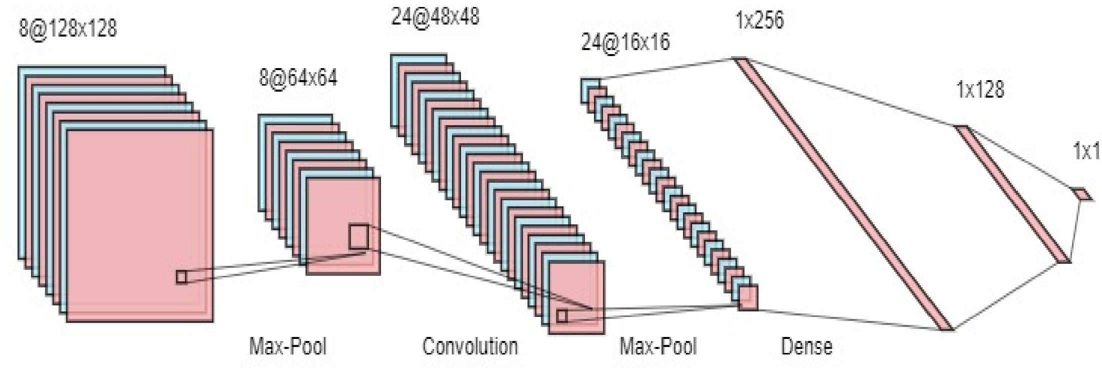


Figure 2.4: General CNN Architecture

> research into convolutional network architectures proceeds so rapidly that a new best architecture for a given benchmark is announced every few weeks to months, rendering it impractical to describe in print the best architecture (8)

What Goodfellow is describing is the topology of individual CNN architectures. Because of the vast reliance on visual data, or humanity's overabundance of visual data, new typologies seem to be published every few weeks. Each with its level of accuracy, localization methodology, and claim to fame. However, there are two some may argue have stood the test of time, at least till now: the YOLO (You-Only-Look-Once) architecture (11; 12; 13; 14), and the Faster RCNN architecture (15). Although, even these come out with new versions almost every other year or so. But in general, most CNN architectures follow a similar format as shown in Figure 2.4

2.1.4.1 The Convolution Operation

The primary function of the convolutional operation within the CNN is to extract features from the input function. While the convolutional operation can be used on other types of grid-like data, this section will focus on the convolutional operations on an image. The convolution function works by combining two functions into one. For use within a CNN this is done by combing the image with a weight matrix, also called a kernel, to produce what is known as a feature map. The convolutional operation can be performed with a weighting function,

$$w(a) \tag{2.3}$$

, where a is the age of measurement. A new function, s, can be obtained by taking the average of the weighted function at every moment, equation 2.4.

$$s(t) = \int x(a)w(t-a)da \tag{2.4}$$

The convolution operation is also commonly written with the asterisk notation see in equation 2.5

$$s(t) = (x * w)(t) \tag{2.5}$$

Within the convolutional operation, x, is referred to as the input function, and the second argument, w, is the kernel. The resulting output, s, is the feature map.

The input variable, x, can be assumed as the image while the kernel, w, is redefined as a filter. The filter is a set of shared weights. These weights can be learned through training the model using a learning algorithm, such as backpropagation. Visually the convolutional operation on an image can be thought of as overlaying the filter onto the image. Additional parameters such as padding and strides are used to obtain the sum of the element-wise products between the image and the filter (16).

2.2 Review of Transfer Learning

Perhaps the main barrier to entry within machine learning comes from a large amount of data and computational power needed to train a model. If a person was to collect several hundreds images of faces, develop a custom CNN architecture, and attempt to train this model on a standard computer the likelihood of success is low. Neural networks require hundreds if not hundreds-thousands of data points to properly generalize and learn the underlying patterns of the model. In addition, even with a standard GPU, this could take days, if not weeks to properly train the model for enough epochs for adequate performance. This is where transfer learning comes in. At a high-level, transfer learning, attempts to take an old model, that has been adequately generalized, and uses this knowledge to retrain the model to learn from a limited dataset. For example, if a model was trained to detect trucks it may be ideal to learn how to recognize trains.

To properly benefit from transfer learning one must have a firm understanding of the initial dataset used. By understanding the lower-level features that the initial dataset possessed one can reap the full benefits of transfer learning. As Dr. Jungme Park et al. state,

> a CNN model learns the image's shapes, edges, and lighting with visual image data in its convolution layers. Because these features are generalized across most types of images, utilizing those learned features from big data in the existing CNN model in a new CNN model with relatively small data samples provides better accuracy than training the new model from scratch. (17)

The idea of transfer learning has formally been around since 1970 (18). Transfer learning can be mathematically defined using the notion of domain and tasks. Using the knowledge gained during the training of source domain D_s, new data can be introduced to remap the outputs for the target domain, D_t.

$$D_s = f_s, P_s(m) \tag{2.6}$$

$$D_t = f_t, P_t(m) \tag{2.7}$$

These domains contain two pieces of vital information, the feature space, f, and the marginal probability distribution $P(m)$. Where m is the set of all data represented within the feature space f, $m = [x_0, x_1, x_2, ..., x_n]^T \epsilon f$ The task, T, each domain is trying to accomplished is defined as $T = y$, $P(c|m)$, where y is the labeled data space, and $P(c|m)$ is a conditional probability distribution. The conditional probability distribution is learned from training the data pairs of xi in the represented data, m, and the y_i label in y. Different techniques are used for transfer learning, depending on the source and the target domain (19).

The practice or implementation of transfer learning typically falls within one of two strategies: feature extracting, and fine-tuning. When using a new model to solve a new task, the goal of the initial model is to extract the low-level features of the new dataset. The CNN layers serve a common purpose of feature extraction; depending on the new task, the layers within the fully connected portion can be retrained by optimizing different hyper-parameters. Although, in time, some may wish to improve the performance of their model. When doing this more data maybe feed through the network in accordance with the same task when this approach is applied the goal is to fine-tune the parameters of the model.

2.3 Review of Semi-Supervised Self-Supervised Learning

Semi-supervised learning has recently grown in popularity among machine learning researchers thanks to its practical relevance in processing unlabeled data of various types. Recent research has shown how S4L can be used on unlabeled text (20; 21), image (22; 23; 24), and even protein sequences (25; 26; 27; 28). There are three main approaches for training within machine learning: supervised (the model is told what the output should be through labeled data), unsupervised (the model decides how to classify the output itself given unlabeled data), and reinforcement learning (the model learns the rules of its world threw positive and negative rewards). Semi-supervised learning falls somewhere in between supervised and unsupervised learning. This approach attempts to combine a large amount of unlabeled data with a small amount of labeled data. Using the labeled data the model will then decide how to categorize the unlabeled data and later use that for further training.

S4L is a special instance of weak supervision. This is a branch within machine learning where the model may be given imprecise data that could be noisy or limited (29). Similar to semi-supervised learning, the goal is still to reduce the need for hard-labeled ground truth data. However, S4L, "take full advantage of the available information in the data and obtain the most accurate prediction" (30). Rather than relying on noisy or limited data, a model using this technique is able to learn from the best foundational features, and then generalize its performance across less ideal datasets.

While in practice, the idea of semi-supervised learning has recently begun picking up in popularity, the idea has been around since the 1960s. Early work in attempting to leverage the predictive labeling and retraining of machine learning models began with a heuristic approach by Dr. Scudder. In this paper, they leverage the central-limit theorem to calculate the approximate probability of error on a pattern-recognition machine, such that it is able to use its outputs as future training examples (31). Within less than a decade, two new approaches were proposed transductive (32) and inductive (33) learning. Transductive learning is fewer ambitions than semi-supervised. Within this framework, the transductive learning model only attempts to predict a classification on an unlabeled dataset. Inductive learning will classify the unlabeled data and if certain conditions have been met append it to the training set. It is common within the literature for semi-supervised learning to be referred to either as transductive or inductive learning given its ability to both classify and determine the correct mapping of inputs to outputs.

The idea of semi-supervised learning can be summarized as the act of gaining the understanding of X such that we can infer the meaning of Y. Authors Chapelle, Scholkopf, and Zien mathematically formalized semi-supervised learning as given a dataset $X = (x_i)_{\epsilon |n|}$ can be split into two parts. $X_l := (x_1, ..., x_l)$, for which labels $Y_l := (y_1, ..., y_l)$ are provided, and the points $X_u := (x_{l+1}, ...x_{l+u})$, the labels of which are not known. With this understanding, it stands to reason the model may only gain useful knowledge from the unlabeled dataset if and only if the unlabeled data carries information useful in the classification of the labeled dataset (34). In other words, the classification of the unlabeled data must exist within the dataset and also be close enough related for the model to understand the relation between the two. For instance, if a model was trained to detect cars, and research applied S4L to the model in an attempt to detect pedestrians, there would be no meaningful improvement. When compared to a model trained on a dataset of hard-labeled pedestrian images.

Using these principles, it sounds as if S4L is set up ideally for the objective of this thesis. Reducing racial bias within pre-existing facial detection systems. If the model is trained on a dataset of human faces, it should understand the key indicators of a face. However, this model was trained with an over-representation of a specific racial group. Therefore, a new unlabeled dataset of uniformly represent racial groups meets the criteria to reap the full benefits of S4L training, and reduce the bias within the system.

CHAPTER III

Literature Review

3.1 Facial Monitoring Pipeline

While developed initially to mitigate distracted driving in traditional vehicles, have proven to be, what some would call, a necessity in level 3 autonomous vehicles. This has to do with the definition of a level 3 autonomous vehicle; the vehicle can operate in most situations; however, drivers may be needed to take over (35). For example, there are many videos online of people leaving the driver's seat of their autonomous vehicle to sleep in the back, read a book, or even make a music video. Unfortunately, these early self-driving cars did not have robust DMS systems integrated to prevent these occurrences. Current DMS systems can be thought of in physiological and facial monitoring domains. Physiological monitoring systems track the vital physiological parameters of the driver to estimate their situational awareness. Pure physiological driver monitoring systems will only track the conditions of the driver and tend to monitor the heart rate and respiratory rate (36; 37). However, this is outside the scope of FMS.

3.1.1 Facial Detection

The first step in any facial monitoring system is to determine the location of the subject's face. Facial detection is a programmatic way to map biometric facial features from a photograph. Understanding where or whose face is being shown is a simple task for a human. One baby can learn within the first few months. However, this task involves several nuances that must be tackled before proper detection can be achieved for a computer.

In the simplest terms, facial detection works by receiving information from a raw photo or video. This basic information is then compared to a known faces or facial features database to understand its environment better. Facial detection in the modern day is accomplished using one, or a combination of, the following methods: Feature Analysis and Neural Networks. Table 1 below shows examples of these methods, their advantages as well as a brief description of their disadvantages.

3.1.1.1 Feature Analysis

The early research effort in facial detection focused on correlating input images with a known database, template matching. However, it became clear that template matching could not generalize to broad edge cases as research progressed. For this reason, a new emphasis was placed on statistical modeling. These models ranged from support vector machines (SVMs) (46) to Random Forest (38), to extracting local feature analysis (45; 44; 43). Regardless of the name, methodology, or complexity, each of these approaches and be broken down into one thing: feature analysis.

A feature is defined as an individual property of the input data. By manually selecting the most significant features within an image, statistical modeling approaches are often better able to learn in higher-dimensional settings such as images. Additionally, removing redundant or useless features makes the training time shorter with fewer memory resources needed. However, feature analysis is considered an NP-hard problem. Several methods solve feature analysis for facial detection; however, each is non-deterministic and, therefore, subject to poor reliability. Additionally, feature analysis methods tend to seek to identify only one solution to a problem, failing to generalize to the level typically required in real-world applications.

3.1.1.2 Neural Networks

More recent efforts in facial detection have exploited the automatic feature analysis capabilities of deep convolutional neural networks (CNN). The convolution function, for which CNNs get their name, is key to this automatic feature analysis. An input image is applied to a convolutional filter, this results in a reduction in the image's dimensions while resting relevant information. As this filter is repeatedly applied to this image it outputs what is called a feature map. This feature map is a low-dimensional representation of the original image focusing on the locations and strength of detected features. Due to the progress made in modern object detections such as Faster R-CNN (47) and YOLO (11) face detection algorithms have been able to generalize and perform to a level unseen with pure feature analysis.

Faster RCNN, known for its single, end-to-end unified network, and use of anchor boxes have seen accurate and quick results in object detection and localization. These attributes make this architecture ideal for face detection. Previous work by Changezheng Zhang et. al. explores the benefits of face detection using the Faster RCNN architecture, which they called FDNet1.0 (48). In their paper, Zhang and team created a light-head Faster RCNN model to improve detection performance and inference speed. This picture uses the ResNet-v 1-101 as a backbone for high-level feature extraction. During training and testing, Zhang and the team resized images to 600, 800, 1000, 1200, and 1400 pixels independently. This resizing allows for minor augmentation. The team used the WIDER FACE dataset (49) for performance evaluation. This dataset was chosen for its, "more challenging collection of images including small scale, illumination, occlusion, background clutter, and extreme poses," (48). Overall the model took 1st place in the easy (95,9% mAP) and medium(94.5% mAP) sets but was only able to take second place in the hard set (87.9% mAP).

Changezheng Zhang et. al.'s new light-head Faster RCNN design was able to display increased performance compared to previous models. However, the authors note there is still room for improvement for faster inference speed and propose designing a light backbone architecture.

YOLO, or "You Only Look Once", is a single neural network for predicting bounding boxes in one evaluation. The YOLO architecture can also be optimized end-to-end and the second and third versions implement the use of bounding boxes. This architecture has many similarities to the previous Faster RCNN architecture; however, YOLO was able to unify classification and bounding box regression training. Additionally, YOLO has been shown to reach a detection speed where the frames per second (FPS) were more than eight times that of Faster RCNN, meaning the model is, while also achieving a higher mean average precision (mAP) (50). In their research Yiheng Zhao, Abdelhamid Mammeri, and Azzedine Boukerche found that they were able to reduce the depth and width of the network backbone, called HeadNet (39). The team also designed a new four-layer based network, called OrganNet, to detect faces with a 96.2% accuracy (39). Due to Yiheng Zhao's team's extended research on estimating the head rotational angle, their team collected primary data using a Hero7 white camera mounted to a vehicle dashboard, as well as the front camera installed on the Alienware 14 laptop for indoor collection efforts resulting in 774 images. The authors note that this is a relatively small dataset, and therefore, there is significant room for further testing on a wider dataset.

By using pre-trained architectures such as Faster RCNN or YOLO, researchers can exploit the proven feature extraction and classification efforts of these architectures. This type of retraining, called transfer learning, is quite useful for detection projects that lack hundreds of thousands of labeled training images. Therefore, many researcher (51; 52; 53; 40; 42) will find a pre-trained backbone such as ResNet (54), Darknet (13), or VGG (55) for feature extraction. While beneficial, the current biggest limitation of transfer learning is negative transfer. Negative transfer is found when the initial target problem is not similar enough to the new detection domain. Additionally, pre-trained models may converge at a local minimum providing similar or worse results than training from scratch.

3.1.2 Landmark Detection

The goal of facial monitoring is to gain insights into the subject's face; however, to do this a computer must be able to understand the location of key landmarks. These landmarks are used to localize important regions of the face such as the eye, mouth, nose, and eyebrows. By understanding the location of such landmarks facial analysis algorithms can perform various tasks to gain insight into the subject. Modern landmark detection systems can be classified into two main categories: holistic and local. Holistic methods can be thought of as reviewing the whole face. As Kim et. al define holistic methods as, "approaches [that] do not require an explicit shape model and landmark detection is directly performed on appearance" (56) Recently Holistic models have begun to experience a surge in regression-based models. For this reason Table 2 below will classify these papers into three sections: Holistic, Local, and Regression-based methodologies. Similarly, local methods are defined by their ability to unify both the appearance and shape of facial landmarks by providing a type of regularization. This section will provide an overview of recent holistic and local landmark detection methods.

3.1.2.1 Holistic Methods

Within landmark detection, holistic methods attempt to leverage a subject's entire face and global shape to determine the location of key facial features. One of the earliest versions of facial landmark detection was developed by Edwards, et al. in 1998 (57). Edwards's team was able to correlate a small number of coefficients to the facial appearance and shape of various subjects using a statistical model. This model was to be called the Active Appearance Model (AAM) and was later improved by Cootes et al (58). The Active Appearance Model can estimate coefficients for landmark detection by performing an interactive calculation of the error image based on the current model's coefficients. While this approach may be able to accurately detect facial landmarks it is no longer considered the optimal solution.

Over the years new, faster, and more accurate holistic methods have been discovered. Joan Albort-i-Medina and Stefonos Zafeiriou discovered the first Bayesian formulation for facial landmark detection, they called it Active Appearance Models (AAMs) (59). By applying a deterministic statistical modeling approach Albort-i-Medina and Zafeiriou were able to outperform discriminatively trained models. Similarly, tzimiropoulos and Pantic also followed a deterministic approach to facial landmark detection. In their approach, the authors argue that Deformable Part Models (DPMs), as one of the most prominent approaches, can benefit by using Gauss-Newton (GN) optimization (60). The unification of these approaches, called the Gauss-Newton Deformable Part Model, was able to significantly reduce the computation cost and outperform current methods.

Recently, a majority of holistic approaches implement some type of regression framework. Yue Wu and Qian Ji cite the rising use of regression models as a third subsection in facial landmark detection. They define this approach as, “implicitly capturing the facial shape and appearance information” (61). Simple machine learning techniques, as in the case of Wimmer et. al (62), Tresadern et. al (63) and Cristinacce (64) have been shown to be several times faster and more accurate than pure statistical models. These methods applied linear and non-linear regression models using features extracted during facial detection as their inputs. More complex machine learning models can also benefit from the use of this feature extraction process such as regression forest (65) and regression trees (66). The feed of facial landmark detection has greatly benefited from the use of machine learning, with the rise in real-time detection and increased performance of GPUs it is believed that regression and more specifically deep learning models will become the standard landmark detection process.

3.1.2.2 Local Methods

Opposite to holistic methods, local methods can detect facial landmarks without the need to fully understand the whole face. By explicitly using these local patches have a slightly better real-time detection performance than traditional holistic models. Authors Jason Saragih, Simon Lusey, and Jeffrey Cohn explore the benefits of deformable model fitting on landmark detection. In this approach, they found that by implying a principled optimization strategy to maximize non-parametric representations through a smoothed estimate hierarchy they were not only able to accurately detect facial landmarks but also robust to partial occlusion (67).

David Cristinacce and Tim Cootes explored the use of the Constrained Local Model (CLM) algorithm. In this approach, the authors used a joint shape and texture appearance model to generate templates for key regions of the face. By doing this the model is able to iteratively compare these templates to the subjects' faces and optimize the shape parameters the maximize the response of the algorithm. By doing so Cristiancce and Cootes showed they were able to outperform the accuracy and robustness of the original AAM search method (68).

3.1.3 Facial Analysis

The advances in computer algorithms and image processing techniques have rendered machine analysis of facial analysis. Facial analysis is a relatively new field in computer vision with many interesting applications from driver monitoring, to security, and even clinical settings. Facial analysis is defined as a "computer system that attempts to automatically analyze and recognize facial motions and facial feature changes from visual information" (69). Facial analysis, however similar to emotion analysis, differs in the implicit level of knowledge required. While facial expressions may convey emotions, the interpretation to understand an actor's intent or feelings is usually also paired with other data sources. These data sources could be body language, tone of voice, situational context, culture, gender, as well as many other factors. This section will focus on the pure analysis of facial features and their importance in facial monitoring systems.

Facial analysis is a key component of facial monitoring systems. Understanding how often a subject is blinking, how wide or how often their mouth is opening can give insight into their level of awareness of a situation. Facial analysis can span far and wide, therefore, this section will limit the review to two key areas used in driver monitoring systems: the Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR). These approaches are key to understanding the attentiveness of a driver. Additionally, these same calculations can be used to understand how drowsy a driver may be.

Researchers explored the benefits of facial landmarks' importance in driver's situational awareness (DSA). Lai proposed the use of the histogram of oriented gradient (HOG) to extract essential features. Lai proposes using a support vector machine (SVM) that feeds the HOG outputs to detect key facial landmarks (70). A crucial part of monitoring the driver's facial features seems to be locating key facial landmarks (51; 42; 52; 53). By using the facial landmarks shown in figure , these authors could determine the eye-aspect ratio (EAR). The EAR compares the eye's vertical and horizontal height to understand if they are open or closed. This information allows the systems to keep track of blink rates to determine the drowsiness of the driver. K. C. Patel et al. showed that by using the ratio between the vertical and horizontal distance of a subject's eye, a machine can estimate the openness of the eye (71). Equation 3.1 expresses the relationship between the height and width of the

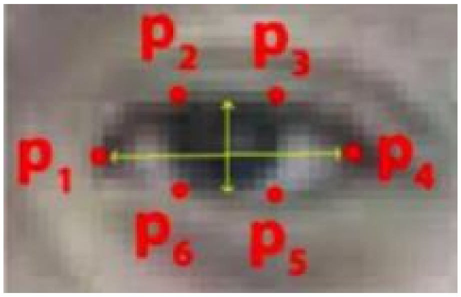


Figure 3.1: Eye Landmarks

eye. The numerator denotes the height, distance between P_2,P_3 and P_6,P_5. While the denominator denotes the width, distance between P_1 and P_4. By expressing the relationship between height and width in such a way a machine is able to understand the following. As the numerator increases the EAR value increases and vice versa when the numerator decreases. This direct relationship is key in understanding many attentiveness calculations. For instance, in driver monitoring systems, if the average EAR value over a set time falls below a threshold value a the computer understands this as a drowsy state (72). However, the rapid decrease and increase in the EAR value caused by normal blinking would not trigger a drowsy state alert. The EAR calculation is a simple, and elegant approach to facial analysis that provides a machine with an exceptional level of situational awareness.

$$EAR = \frac{|P_2 - P_6| + |P_3 - P_5|}{2 *_1 -P_4|} \tag{3.1}$$

Similarly, the mouth aspect ratio (MAR) measures how open the observer's mouth is. Chadiwala and Agarwal used the MAR ratio to monitor how often the driver is yawning to gauge drowsiness (42). Facial landmark detection has shown to be a beneficial way to monitor a driver's attention. Researchers have taken the spirit of facial landmark detection and trained machine learning models to understand the DSA levels (70; 38; 48; 73; 40; 39). Schwarza et al. explored the benefits of combining facial landmarks detection with the random forest algorithm. This research showed that using the facial landmarks to estimate eye closure, eye gaze, facial orientation, and distance between the nose and mouth, the random forest algorithm accurately predicted DSA roughly 99% of the time (38). Convolutional neural networks have also begun to be employed in DMS (48; 73; 40; 39). The use of convolutional neural networks allows the system to better generalize to edge cases. The facial landmark points of interest used in calculating the mouth aspect ratio are shown in figure above (74). Similar to the EAR calculation, the MAR formula, equation 3.2, expresses the relationship between the vertical and horizontal distances of the mouth. Again, this relationship allows the computer to understand that as the vertical distance increases the mouth is left open. Insights can be made about the number of times an actor's mouth is opened wider than a set threshold, this would allow a machine to understand the person is yawning and therefore, may be drowsy. This is also a common approach to monitoring drivers.

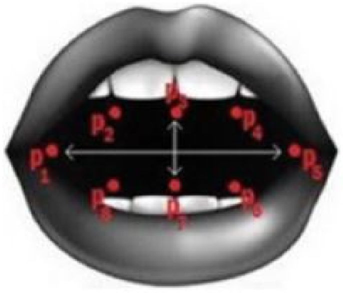


Figure 3.2: Mouth Landmarks

$$MAR = \frac{|P_2 - P_8| + |P_3 - P_7| + |P_4 - P_6|}{2 * |P_1 - P_5|} \tag{3.2}$$

3.2 Semi-Supervised Self-Supervised Learning

Semi-supervised self-supervised learning (S4L) is an approach developed to tackle the problem of semi-supervised learning on image classifiers. Semi-supervised learning techniques attempt to learn from both labeled and unlabeled data. While Self-supervised learning describes a framework of data labeled using only unsupervised data. S4L is the unification of these two approaches into one; however, in literature, an S4L is commonly referred to as semi-supervised learning. Proposed by Dr. Xiaohua Zhai et al. at Google Research, Brain Team in 2019 attempted to train a single network originally to predict the rotation applied to both labeled and unlabeled images (22). This paper describes a three-stage approach to training:

- Train a semi-supervised model
- Use the model to produce pseudo labels for all images within the dataset
- using these pseudo labels retrain the same model to predict these labels

Building off these three basic steps the Google Research, Brain Team in collaboration with Carnegie Mellon University published a paper on iterative self-training (75). Building off the idea of semi-supervised training, these authors were able to achieve 84% top-1 accuracy on ImageNet, over 2% better than current state-of-the-art models. Additionally, by using semi-supervised approaches, the authors did not require the typical 3.5 billion weakly labeled images used with current models. To achieve these results, the researchers trained an EfficientNet model explicitly on the labeled ImageNet images (75). This model will become the teacher to decide the label for over 300 million unlabeled images. These pseudo-labels generated by the teacher are then fed as training examples for a "student" model. After training the student becomes the teacher, and the process continues.

Similarly, researchers at Apple explored the quality of S4L data labeling (76). In this research, the authors hoped to understand the efficiencies and S4L labeling while verifying the annotations under a human-in-the-loop setting. Through this they showcased, "that the latest advancements in the field of self-supervised visual representation learning can lead to tools and methods that benefit the curation and engineering of natural image datasets, reducing annotation cost and increasing annotation quality" (76).

These three ideas (self-labeling, iterative training, and human-in-the-loop verification) are the north star for this thesis research. This thesis hopes to explore the possibility of using the iterative S4L techniques to reduce the racial bias of preexisting networks through a small set of manually selected unlabeled images.

3.3 Bias within Deep Learning Systems

Artificial intelligence is nothing more than a generalization of what it is taught. If a model is taught to only detect red cars, that model will only detect red cars, they are not objective. When developing these models it is the engineer or researcher's responsibility to be aware of the bias that can occur. These biases could be in part because of the access or representation of data, or unconscious decisions made by the developer. There are hundreds of different types of cognitive biases that have the potential to skew a model's prediction. Google talks about five of these biases: reporting, automation, selection, group attribution, and implicit bias (77).

Of these, three are what this thesis will expand on. Automation bias is defined as, "a tendency to favor results generated by automated systems over those generated by non-automated systems, irrespective of the error rates of each" (77). Through the use of semi-supervised learning, one opens their model to be vulnerable to this type of bias. Therefore, this thesis argues that a human must always be involved in some way in the selection of data used for training. Selection bias is caused by a dataset that is not reflective of the real-world distribution. This thesis builds off the already-known issue of selection bias within the preexisting benchmark dataset. This thesis also acknowledges the relative cost associated with the collection, annotation, and deployment of a new benchmark dataset, and hopes to provide a solution that can rectify the effects of selection bias within preexisting systems. Finally, group attribute bias is the generalization of what is true of individuals to an entire group (77). Within preexisting benchmark datasets, specifically, face detection datasets, data is labeled as a binary. Is there a face, or is there not a face? However, a more granular understanding of the data is needed to truly understand the model's performance and avoid the in-group homogeneity bias.

Table 3.1: Facial Detection Comparison Table

Reference ID	Methods	Advantages	Disadvantages
(38)	Feature Analysis	• Low computational Cost	• Low Accuracy • Scalability
(39)	Neural Networks- Transfer Learning	• High generalization • low computation cost for retraining YOLOv3 model	• Low Dataset size (774 images)
(40)	Neural Networks	• Generalizes well to new data • High Accuracy	• High computational cost • High overhead
(41)	Feature Analysis and Neural Networks	• Generalizes well to new data • High Accuracy	• High computational cost • High overhead
(42)	Feature Analysis and Neural Networks	• Generalizes well to new data • High Accuracy	• High computational cost • High overhead
(43)	Feature Analysis	• Low computational Cost	• Fails to generalize
(44)	Feature Analysis	• High recognition rate • Image quality does not need to be ideal • Low computational cost	• Poor run-time for detection
(45)	Feature Analysis	• works on low-resolution images • Works on binary images	• average performance metric 85% accuracy

CHAPTER IV

Review of Faster RCNN Architecture

The Faster R-CNN architecture is the third iteration of a series of region-based convolutional neural networks. Region-based convolutional neural networks are simply neural networks that are not only able to classify objects, but also locate these objects within an image. Traditionally, object classification and localization can be computationally expensive, and therefore, not exactly suitable for real-time detection. However, Faster R-CNN was able to reduce the region proposal bottleneck, therefore reducing time, thanks to two main contributions: the Region Proposal Network (RPN) and its use of anchor boxes (47).

The Faster R-CNN architecture is networks. The first is a fully connected convolutional network. The goal of this first network is to reduce the dimensionality of the input images and produce the feature map. Once this feature map is created potential regions of interest are generated. The second network also produces a range of regions of interest. This network is called the RPN, Region Proposal Network. The RPN is modeled as a fully convolutional network whose input is the feature map of the previous layer and output is a set of rectangular object proposals, each with a confidence score. These proposed regions are generated using a sliding window technique (78). The outputs of each window are fed into what the authors describe as, "two siblings fully-connected layers-a box-regression layer (reg) and a box-classification layer (cls)"

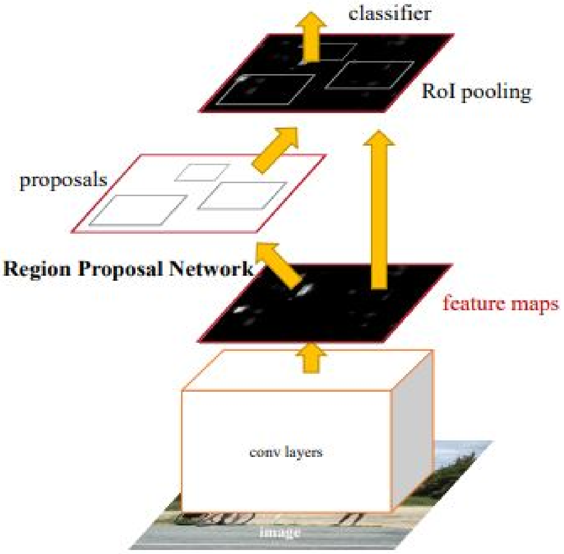


Figure 4.1: Faster R-CNN Unified Network

(47). Finally, the model can review and pool all regions of interest and filter them into the classification layer. This is where the final output can be seen. Figure 4.1 is taken from the original Faster R-CNN paper, as it provides a visual depiction of how these networks are connected into a single unified model (47).

So what allows the Faster R-CNN architecture to operate near real-time, is the use of the ROI Pooling layer. Rather than producing individual calculations for each proposal independently, and repeatedly, the architecture can leverage a single calculation multiple times. Additionally, the use of anchor boxes seems to have provided the model with additional support. An anchor is the central point of the sliding window. Each sliding window simultaneously predicts multiple region proposals called anchors. The anchor or proposed region is then passed to the reg, regression, and cls, classifier, layers. The reg layer in Figure 4.2, again from the original Faster R-CNN paper (47), is used to output the coordinates of the anchors while the cls layer estimates the object's probability. An essential feature of the approach is the anchor translation invariant,

> if one translates an object in an image, the proposal should translate, and

the same function should be able to predict the proposal in either location (16)

In other words, for any image, the scale and aspect ratio can be applied to the network. The aspect ratio of an image is the width of the image divided by the height of the image, while the scale is the image's overall size. Three scales and three aspect ratios were chosen by the developers resulting in a total of nine possible proposals for each pixel. This is how the value k, the number of anchors, is decided.

The multi-scale anchors proposed in this RPN algorithm result in a "Pyramid of Anchors" instead of a "Pyramid of Filters". The change from a pyramid of filters to a pyramid of anchors resulted in a more cost-efficient and faster time than previous algorithms. Anchors are assigned labels based on two factors. In testing the anchor with an Intersection-Over-Union overlap higher than 0.7, and during training the anchors with the highest Intersection-over-union overlap with the ground truth box are used.

The Faster R-CNN architecture is arguably showing its age. First published in 2016, this architecture is more than a few years old now. However, nonetheless, it

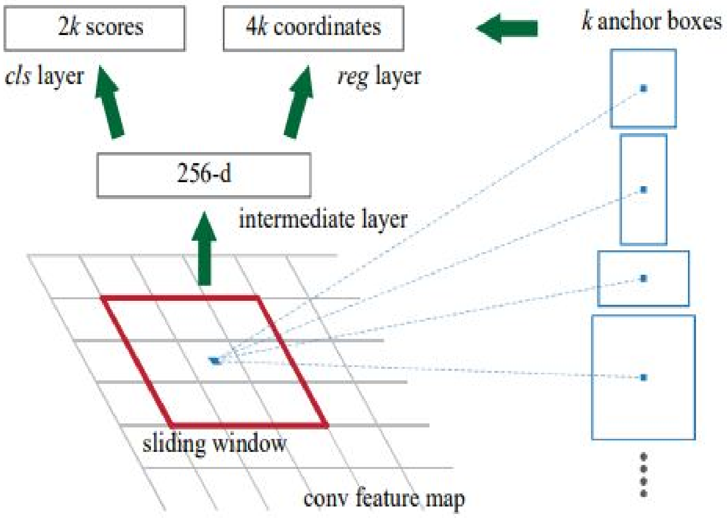


Figure 4.2: Region Proposal Network

has stood the test of an AI development cycle still being regarded as one of the ideal detection models, and still used commonly in recent publications on transfer learning (79; 80; 81; 82; 83).

For this thesis, the Faster R-CNN architecture was chosen over other architectures for three main reasons. First, the model has been shown to achieve real-time detection. Not only, but the training time required for S4L seemed comparable to the timeline of the project. Secondly, the architecture was initially trained on the COCO (84). Finally, previous research has shown that the Faster R-CNN model can be retrained, using transfer learning, to detect human faces (48). The understanding that a model has already proven to be capable to detect faces, removed the question of if this thesis could leverage transfer learning for training the initial model.

CHAPTER V

Review of Datasets

If data is considered gold in the 21st century, quality data is the new oil. There's a belief in the AI community that a larger quantity of data allows for better generalization of a model. As the old saying goes, ”garbage in garbage out”. Data itself can be biased, a researcher could have access to tens of millions of data points, but could misrepresent the actual population.

5.1 Review of Unbalanced Benchmark Datasets

Dr. Joy Boulamwini and her team conducted a review of some benchmark datasets for face detection with MIT back in 2018 (3). Two main datasets shown in figure 5.1, from Dr. Boulamwini's findings denote the discrepancies between unbalanced. The first dataset reviewed is the Adience gender classification benchmark. This dataset was released in 2014, containing 2284 individuals (85). the IARPA Janus Benchmark A (IJB-A) dataset was developed specifically to, ”augment more challenges to the face recognition task” which contains a wide variety of poses, illumination, expressions, and resolutions. It contains 5712 images, and over 2000 videos with 500 identities (86).

As Figure 5.1 shows there is a clear discrepancy in the distribution of dark females and lighter males. A simple inference of the data would suggest that a model trained

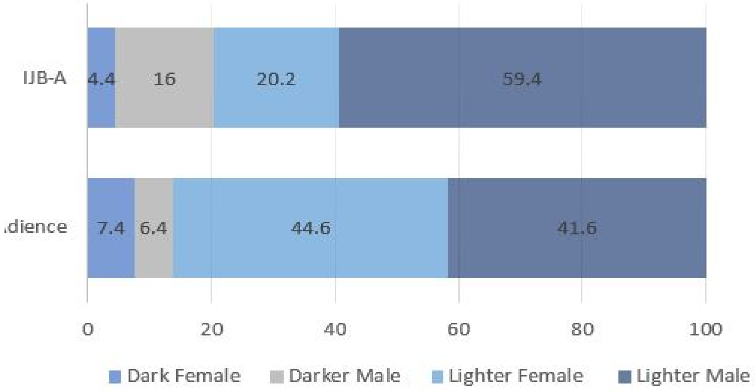


Figure 5.1: Gender Shades Comparison of Benchmark Datasets

with these images would have a bias toward detecting lighter males. Dr. Buolamwini went on to test this hypothesis on three industry models, Face++, Microsoft's Cognitive Service Face API, and IBM's Watson Visual Recognition API (87; 88; 89). It was found that at a minimum models that were trained with unbalanced datasets had a discrepancy of 23.8% in their positive predictive values between lighter males and darker females. At a maximum this discrepancy was 36% (3). For this reason, Dr. Buolamwini argued for the development of a more balanced dataset.

5.2 Review of Balanced Benchmark Datasets

As part of Dr. Buolamwini's research on understanding the effects of gender and race distribution within benchmark datasets, she published her balanced benchmark dataset. This dataset titled the Pilot Parliaments Benchmark (PPB) is comprised of 1270 individuals from 6 countries' national parliaments: Rwanda, Senegal, South African, Ice-land, Finland, and Sweden (3). The gender classification was selected using the binary sex labels of female and male, while skin pigmentation was determined using the Fitzpatrick six-point labeling system (90). Figure 5.2 shows the distribution of the PPB dataset.

This dataset has a major problem that is not accounted for in its distribution. The

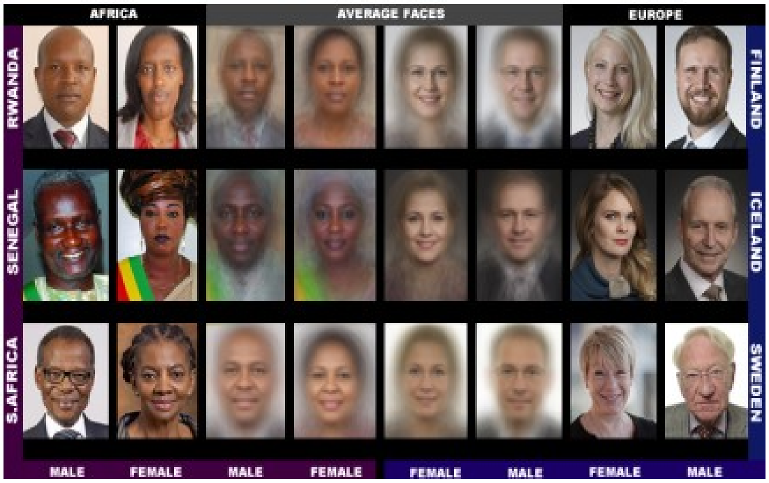


Figure 5.2: Pilot Parliaments Benchmark Distribution

data was collected using by mining the photos from government websites, therefore limiting the wide variety of poses, illumination, expressions, and resolutions found in unbalanced benchmark datasets. While a model trained on this dataset may be more uniform in its detection of different racial groups, it can be assumed the model will be unable to generalize to real-world situations.

Similarly, other balanced datasets have been developed such as the FairFace dataset (91). The FairFace dataset developed in 2019, contains 108501 images of 7 different racial groups: White, Black, Indian, East Asian, Southeast Asia, Middle East, and Latino collected from the YFCC-100M Flicker dataset (92). Much like the PPB dataset, the FairFace dataset has major drawbacks when compared to traditional benchmark datasets. The FairFace dataset does have variations in poses and illumination; however, all images are cropped to isolate the face of the subject.

To the best of the author's knowledge, these are the only two uniform face datasets available for public use. As it's been described while these datasets are balanced regarding race, they are unbalanced to deal with real-world scenarios. Therefore, it would be unadvised to use either of these datasets solely to train a face detection model for real-world use. However, these datasets may be ideal for reducing bias in pre-trained face detection systems. This thesis will explore the effects the FairFace

Figure 5.3: WIDER FACE Dataset Examples

dataset will have on reducing bias within a pre-trained face detection system.

5.3 WIDER FACE Dataset

Within this thesis, the WIDER FACE dataset will be used for the initial transfer learning of the Faster R-CNN model. This dataset was originally published in 2015 containing 393,000 faces. These faces included individuals, groups, and actives, and are overall fairly representative of the real-world environment (49). This dataset was chosen for large annotation size and rich diversity in occlusion, poses, event categories, and face bounding boxes. Example images from this dataset can be seen in figure 5.3. However, most importantly, this dataset was chosen done on the work of Dr. Karkkainen and Dr. Joo at the University of California, Los Angeles, who published their findings that the WIDER FACE and other benchmark datasets are biased towards lighter skin faces by around 80% (91).

Using a known bias dataset, with high variation, provides this thesis the ability to train a model to be representative of a real-world industry model. The WIDER FACE dataset is split into a 40-10-50 training, validation, and testing dataset with bounding boxes from 10-300 pixels.

Within this thesis, the total training set will be used to conduct transfer learning on the Faster R-CNN model. For better generalization, the training set will be

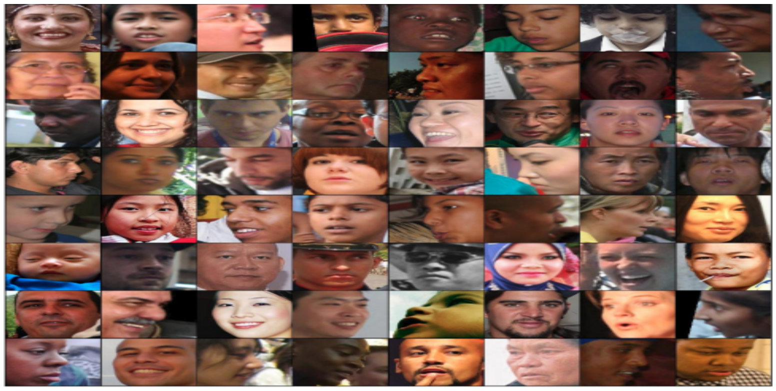


Figure 5.4: FairFace Dataset Examples

rotated between -15 to 15 degrees. The level of variation in the data along with the generalization of image augmentation should emulate the real-world environment.

5.4 FairFace Dataset

The authors of the FairFace dataset argue that the granular nature of the Pilot Parliaments Benchmark dataset's that annotation based on skin pigmentation can be heavily affected by illumination of lighting conditions. For this reason, the authors labeled the FairFace dataset according to seven race groups based on the U.S. Census Bureau: White, Black, Indian, East Asian, Southeast Asian, Middle East, and Latin. The authors also note that "Race and ethnicity are different categorizations of humans. The race is defined based on physical traits and ethnicity is based on cultural similarities" (91). Additionally, they go on to state, "Latino is often treated as an ethnicity, but we consider Latino a race, which can be judged from the facial appearance" (91). Finally, it is important to note that labeling based on one's interpretation of physical appearance is subject to bias. More work must be done to create a more balanced dataset, not just in the distribution of represented racial groups, but a more granular representation of the subject's identity. Images within this dataset have slight diversity in pose and illumination; however, all images have been cropped to isolate a single subject's face. These images also contain no bounding box information but rather ground truth labeling on the age, gender, and race of the individual. of the 108,000 images, only the first 100 images for each race group will be used for S4L training, a total of 700 S4L training images. Again these images will be augmented by rotating between -15 to 15 degrees.

This thesis aims to understand the effects of semi-supervised self-supervised learning on bias reduction through training on the Faster R-CNN by use of the 700 hand-selected FairFace images.

CHAPTER VI

Semi-Supervised Self-Supervised Training with Faster RCNN

6.1 Training Environment

6.1.1 Computing Setup

As part of retraining the Faster R-CNN architecture, an ideal training environment must be discussed. For retraining a model of this size, with such a large variety of data there are two main hardware components to consider. The amount of Random-Access Memory (RAM) and the quality of the Graphics Processing Unit (GPU). Both are used for memory access and speeding up the training process. In theory, subpar hardware could be used to train the network; however, given the cost to performance reward, this would be ill-advised. The amount of time saved by high-performing hardware allows for more training cycles, and therefore, more time to optimize hyperparameters.

The author of this thesis had access to a Dell G5 with 16GB RAM and a 2070 NVIDIA GeForce RTX GPU. However, This computer proved to be ill-fitted for the task as a single training cycle took over a week to complete. Therefore the decision was made to purchase a new Dell G5 with 32GB Ram and an NVIDIA GeForce RTX 3070 GPU. Table 6.1.1 shows the complete hardware setup used for training.

The decision to buy a whole new Dell G5 computer had to do with the cost comparison of the 3070 GPU and the pre-built system. At the time 3070 GPUS were selling for over $2500 while the Dell G5 was selling for around $1500.

Operating System	CPU	RAM	Graphics Card
Windows 11	Intel i7	32GB DDR3 2666 MHz	NVIDIA GeForce RTX 3070

Table 6.1: Dell G5 Hardware Setup

This training setup has provided a much better environment than the original. Not only did it provide more memory and a better GPU. The updated Dell G5 was able to reduce the training time of a single cycle from over a week to three days.

6.1.2 Transfer Learning Setup

As with most transfer learning projects the first in setting up the training environment is formatting the data in a way the model can understand. For the transfer learning portion of this thesis, only the WIDER FACE training dataset was used. The Faster R-CNN architecture was trained using the MATLAB toolbox. For the Faster R-CNN architecture ground truth bounding boxes are stored as (x min, y min, x max, y max) and the class label. Luckily for this research, the WIDER FACE dataset is already annotated in this format. Therefore, no prepossessing of the images was done.

After it was determined the annotations are in the correct format, augmentation can begin. Data augmentation is typically used to improve the robustness and generalization of a deep learning model. Often augmentation is applied to artificially increase the dataset size. As it's been stated the only augmentation applied within this research is rotational between -15 and +15 degrees.

With all the data in the correct format, training, through transfer learning, can finally begin. One could simply define random variables for the hyperparameters and blindly accept the output. However, that is not research. Within this thesis, the model was first trained multiple times using various learning rates for 0.1 to 0.0001 degrading by a magnitude of 10 at each step. It was noted that a majority of the learning rates failed to train the model correctly. At both extremes the model never seemed to converge, causing erratic updates of the weight functions.

It was found that a learning rate of 0.001 was able to produce the most stable results and avoid local minimums while retraining the model. Furthermore, to prevent overfitting, the L2 regularization technique was used. The model was trained using stochastic gradient descent with a learning momentum of 0.9.

After finding the ideal hyperparameters the model was trained for a total of 10 epochs, saving the model weights at each epoch. The epoch is a hyperparameter that defines the number of times each data point within the training set is evaluated on the model. It was found that the final epoch produced the highest level of confidence on the WIDER FACE Test dataset.

Traditionally, in object detection and localization models, the performance is based on the Mean Average Precision (mAP). However, because the FairFace dataset does not include ground truth annotation this would be a misleading performance metric when comparing the initial and final versions of the model.

The results of the transfer learning step will be further discussed in section 6.2.

6.1.3 Semi-Supervised Self-Supervised Setup

To properly benefit from semi-supervised training with unlabeled data, two assumptions must be met: smoothness and cluster. The smoothness assumptions states that, "if two points x_1,x_2 in a high-density region are close, then so should be the corresponding outputs y_1, y_2" (34). In other words, the unlabeled data should be closely enough related to the original labeled training data, such that the model is able to find a relative relationship between them. Given that labeled and unlabeled data are both face images, within the RGB color scheme one can say the smoothness assumptions have been passed.

With the understanding that the smoothness assumptions have been met, the data needs to be checked for the clustering assumptions. The clustering assumption states that "if points are in the same cluster, they are likely to be of the same class" (34). This assumption seems to apply more to multi-classification networks. Within this case, because all 700 unlabeled images contain a face, and the initial model was trained only to detect faces, one can assume the clustering assumption has been met. However, if the model was trained to detect people and then S4L was used to learn to detect faces, the validity of passing the clustering assumption becomes more ambiguous.

Finally, passing both the smoothness and clustering assumptions of semi-supervised learning, this thesis can assume moving forward that the model will be able to fully benefit from S4L training.

After the initial model has been retrained through transfer learning, this thesis will implement the teacher-student approach of S4L. In this approach, the initial model can be thought of as a teacher, who will define pseudo-labels to the 700 hand-selected images of the FairFace dataset. These pseudo-labels will then be used to retrain the model, or student, and update its weights. The student will grow up to become a teacher and educate the next generation. This process was repeated for a total of two-generation. Figure 6.1 provides a graphical representation of this generational training process. S4L can be thought of as gaining generational knowledge. Each generation is expected to pass on its knowledge to the next generation while the new generation is expected to learn and become better than their ancestors. For this research, it was decided a detector confidence level above 70% will be saved as pseudo-labels and used to retrain the model.

During the S4L training cycle, the research has the opportunity to redefine hy-

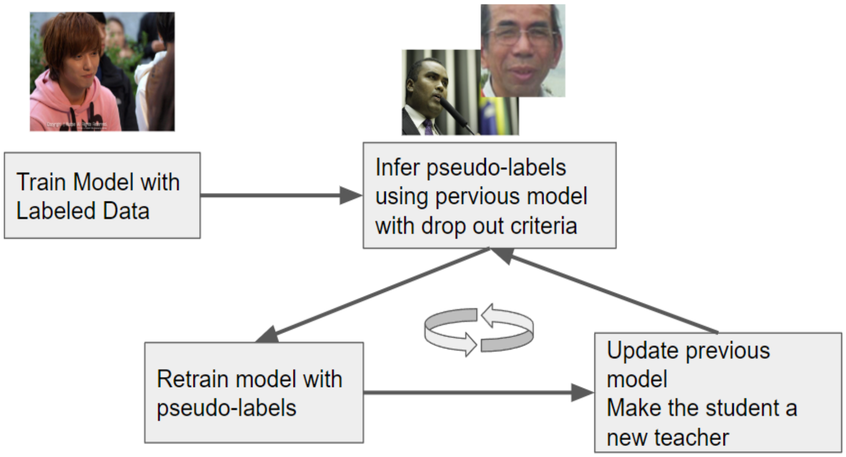

Train Model with Labeled Data
Infer pseudo-labels using pervious model with drop out criteria
Retrain model with pseudo-labels
Update previous model Make the student a new teacher

Figure 6.1: Semi-Supervised Self-Supervised Generational Training Process

perparameters. In some cases, this may be beneficial. However, for this thesis, it was found that using the defined hyperparameters using during the transfer learning setting still provided an adequate training environment.

6.2 Results Analysis

Again it's important to state, traditionally, the mean average precision score (mAP) is used to understand the performance of an object detection model. The mAP correlates to the amount of overlap between the predicted and ground truth bounding box and the model's ability to correctly identify all positive examples. The FairFace dataset, however, does not contain ground truth bounding box information. The use of pseudo-labels for training is one thing, although, becomes redundant for the mAP performance metric. Additionally, this would be ill-advised as the model would be determining its ground truth labels and testing against itself.

For this reason, the model's average confidence score for each racial group will be used as the performance metric. The use of the confidence score as a performance metric is also better for the real-world environment in which this application could be used. The goal of this technique is to not only reduce racial bias but reduce

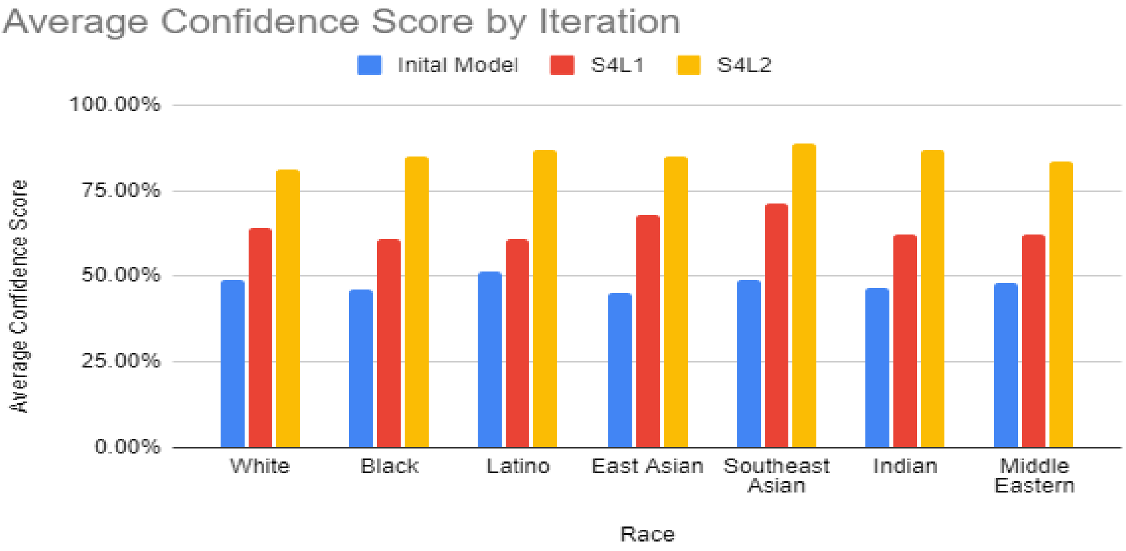


Figure 6.2: Semi-Supervised Self-Supervised Generational Training Process

the cost associated with the annotation. The tracking of confidence across classes or subcategories attributes would provide a better insight into how the model is perceiving and understanding its environment.

The Faster R-CNN was originally trained on the MS COCO dataset, and therefore able to detect over 90 different classes. Unfortunately, of these 90 classes, human faces were not one of them. For this reason, transfer learning must be utilized for this thesis.

The Faster R-CNN architecture was retrained using the WIDER FACE training dataset. After this, the performance of the model was determined using the WIDER FACE test dataset. The model was able to reach over 80% average confidence on this dataset. This is not ideal for detection in the real world. However, it is a great starting point for semi-supervised self-supervised learning to build off of.

The retrained Faster R-CNN model was also tested on the full FairFace dataset. The model ended up performing extremely low, roughly 40% confidence, on all racial groups. This can be seen in figure 6.2, followed by the results of the following two S4L generations.

This graph shows a clear reduction in bias across all seven different racial groups as the model was able to correct itself through S4L training. The model was able to double its confidence in detecting all racial groups. By only giving the model access to 700 hand-selected, unlabeled images that reflected a known bias in the system, semi-supervised self-supervised training was able to reduce racial bias significantly from the poor performing face detection system.

Model	Mean	Standard Deviation
Amazon	0.941	0.03
Microsoft	0.806	0.042
Face++	0.896	0.066
IBM	0.9	0.061
S4L	0.868	0.024

Table 6.2: Industry Model Performance on FairFace Test Dataset

Table 6.2 shows the results of four industry facial detection models tested on the FairFace dataset and the results of the final iteration of semi-supervised self-supervised training. The results of these four models: Amazon, Microsoft, Face++, and IBM, were obtained for the original paper, *FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age for Bias Measurement and Mitigation* (91). This table shows the average accuracy, mean, of each model as well as their standard deviation. While the S4L model unperformed all except for Microsoft on the FairFace test set, it is more important to review the standard deviation. The standard deviation explains how spread out the accuracy across different racial groups is spread across the mean. The mean accuracy of the model hides it's racial bias. While a model may achieve an average accuracy of 90% this could hid the fact that white individuals are detected at 99% of the time while black individuals are detected at 70% of the time.

The standard deviation gives illuminates an idea of the level of disparity between the highest detection and lowest detection rate. The goal of this research is create a more equal level of detection across all seven racial groups.

This thesis was able to archive the highest level of clustering around the mean, or lowest standard deviation our when compared to all four industry models. It is important to note a potential limitation in this research. When designing the S4L model only 700 training images were used, this could possibly have negatively affected the models average level of detection. It is believed that if the research was to be redone the full training dataset should be used, this could increase the average accuracy while also further reducing the standard deviation.

CHAPTER VII

Conclusion and Future Work

This research has shown that through the selection of unlabeled images that reflect a known bias within a system, semi-supervised self-supervised learning holds the key to significantly improving the confidence of a facial detection model. There is an important distinction to be made about leveraging the independence of semi-supervised self-supervised learning and human intervention. While this approach was able to reduce the need for manual data annotation, therefore reducing associated financial costs. A human-in-the-loop is critical in the selection of data. At this point, there is no way for a model itself to understand its biases and select useful data automatically in a meaningful way. Additionally, it is unadvised for a black box of this size to exist compounding biases could exist and go unnoticed. By reducing the need for manual annotation, the human-in-the-loop can put forth more meaningful effort in the selection of training data. The selection and representation of data, also known as the quality, seems to play a more important role in reducing bias than the quantity of data. This work should be further explored to formally generalize this methodology to understand how S4L training can reduce bias in all deep learning systems. By reducing the need for manual data annotation this methodology can significantly reduce the cost and time needed to develop more robust autonomous systems.

BIBLIOGRAPHY

[1] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[2] R. Max. (2021) Facebook apologizes after a.i. puts 'primates' label on video of black men. [Online]. Available: https://www.nytimes.com/2021/09/03/technology/facebook-ai-race-primates.html

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

[4] T. Oloko. (2016) What snapchat not recognizing my face can teach us about building inclusive products. [Online]. Available: https://www.linkedin.com/pulse/what-snapchat-recognizing-my-face-can-teach-us-building-toni-oloko/

[5] J. G. Asare. (2020) Does tiktok have a race problem? [Online]. Available: https://www.forbes.com/sites/janicegassam/2020/04/14/does-tiktok-have-a-race-problem/?sh=4033e3183260

[6] J. Pesenti. (2021) An update on our use of face recognition. [Online]. Available: https://about.fb.com/news/2021/11/update-on-use-of-face-recognition/

[7] A. Addagatla, "A study of artificial neural networks (ann)," 2020.

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* MIT press, 2016.

[9] W. Mcculloch and W. Pitts, "A logical calculus of ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 127–147, 1943.

[10] M. Minsky and S. Papert, "Perceptrons: An introduction to computational geometry (expanded edn)," 1988.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[12] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[13] J. Redmon and a. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[14] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[15] H. Jiang and E. Learned-Miller, "Face detection with the faster r-cnn," in *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)*. IEEE, 2017, pp. 650–657.

[16] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[17] J. Park, W. Yu, P. Aryal, and V. Ciroski, "Comparative study on transfer learning for object classification and detection," in *AI-enabled Technologies for Autonomous and Connected Vehicles*. Springer, 2023, pp. 125–142.

[18] S. Bozinovski and A. Fulgosi, "The influence of pattern similarity and transfer learning upon training of a base perceptron b2," in *Proceedings of Symposium Informatica*, vol. 3, 1976, pp. 121–126.

[19] J. Park, "Lecture on convolution neural network," 2019.

[20] R. Johnson and T. Zhang, "A high-performance semi-supervised learning method for text chunking," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 2005, pp. 1–9.

[21] P. Izmailov, P. Kirichenko, M. Finzi, and A. G. Wilson, "Semi-supervised learning with normalizing flows," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4615–4630.

[22] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4l: Self-supervised semi-supervised learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1476–1485.

[23] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-supervised learning with deep generative models," *Advances in neural information processing systems*, vol. 27, 2014.

[24] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *Advances in neural information processing systems*, vol. 32, 2019.

[25] T.-P. Nguyen and T.-B. Ho, "Detecting disease genes based on semi-supervised learning and protein–protein interaction networks," *Artificial intelligence in medicine*, vol. 54, no. 1, pp. 63–71, 2012.

[26] B. R. King and C. Guda, “Semi-supervised learning for classification of protein sequence data,” *Scientific Programming*, vol. 16, no. 1, pp. 5–29, 2008.

[27] L. Käll, J. D. Canterbury, J. Weston, W. S. Noble, and M. J. MacCoss, “Semi-supervised learning for peptide identification from shotgun proteomics datasets,” *Nature methods*, vol. 4, no. 11, pp. 923–925, 2007.

[28] L. Moffat and D. T. Jones, “Increasing the accuracy of single sequence prediction methods using a deep semi-supervised learning framework,” *Bioinformatics*, vol. 37, no. 21, pp. 3744–3751, 2021.

[29] T. Blog, “Weak supervision: A new programming paradigm for machine learning,” 2022.

[30] N. Al-Azzam and I. Shatnawi, “Comparing supervised and semi-supervised machine learning models on diagnosing breast cancer,” *Annals of Medicine and Surgery*, vol. 62, pp. 53–64, 2021.

[31] H. Scudder, “Probability of error of some adaptive pattern-recognition machines,” *IEEE Transactions on Information Theory*, vol. 11, no. 3, pp. 363–371, 1965.

[32] V. Vapnik and A. Chervonenkis, “Theory of pattern recognition,” 1974.

[33] M. G. Madden, “Hierarchically structured inductive learning for fault diagnosis,” *WIT Transactions on Information and Communication Technologies*, vol. 20, 1970.

[34] O. Chapelle, B. Scholkopf, and A. Zien, “Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews],” *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[35] J. Shuttleworth, “Sae standards news: J3016 automated-driving graphic update,” 2019.

[36] K. Bylykbashi, E. Qafzezi, M. Ikeda, K. Matsuo, and L. Barolli, “Fuzzy-based driver monitoring system (fdms): Implementation of two intelligent fdmss and a testbed for safe driving in vanets,” *Future Generation Computer Systems*, vol. 105, pp. 665–674, 2020.

[37] H. Rahman, S. Barua, and B. Shahina, “Intelligent driver monitoring based on physiological sensor signals: Application using camera,” in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 2637–2642.

[38] C. Schwarz, J. Gaspar, T. Miller, and R. Yousefian, “The detection of drowsiness using a driver monitoring system,” *Traffic injury prevention*, vol. 20, no. sup1, pp. S157–S161, 2019.

[39] Y. Zhao, A. Mammeri, and A. Boukerche, "A novel real-time driver monitoring system based on deep convolutional neural network," in *2019 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, 2019, pp. 1–7.

[40] D. Yang, X. Li, X. Dai, R. Zhang, L. Qi, W. Zhang, and Z. Jiang, "All in one network for driver attention monitoring," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 2258–2262.

[41] S. Shaily, S. Krishnan, S. Natarajan, and S. Periyasamy, "Smart driver monitoring system," *Multimedia Tools and Applications*, vol. 80, no. 17, pp. 25 633–25 648, 2021.

[42] J. Chandiwala and S. Agarwal, "Driver's real-time drowsiness detection using adaptable eye aspect ratio and smart alarm system," in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1. IEEE, 2021, pp. 1350–1355.

[43] F. Jiao, W. Gao, X. Chen, G. Cui, and S. Shan, "A face recognition method based on local feature analysis," in *Proc. of the 5th Asian Conference on Computer Vision*. Citeseer, 2002, pp. 188–192.

[44] J. Ng and H. Cheung, "Dynamic local feature analysis for face recognition," in *International Conference on Biometric Authentication*. Springer, 2004, pp. 234–240.

[45] M. A. Talab, S. Awang, and M. D. Ansari, "A novel statistical feature analysis-based global and local method for face recognition," *International Journal of Optics*, vol. 2020, 2020.

[46] P. Shih and C. Liu, "Face detection using discriminating feature analysis and support vector machine," *Pattern Recognition*, vol. 39, no. 2, pp. 260–276, 2006.

[47] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[48] C. Zhang, X. Xu, and D. Tu, "Face detection using improved faster rcnn," *arXiv preprint arXiv:1802.02142*, 2018.

[49] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[50] L. Tan, T. Huangfu, L. Wu, and W. Chen, "Comparison of yolo v3, faster r-cnn, and ssd for real-time pill identification," 2021.

[51] S. Ceamanunkul and S. Chawla, "Drowsiness detection using facial emotions and eye aspect ratios," in *2020 24th International Computer Science and Engineering Conference (ICSEC)*. IEEE, 2020, pp. 1–4.

[52] S. Sathasivam, A. K. Mahamad, S. Saon, A. Sidek, M. M. Som, and H. A. Ameen, "Drowsiness detection system using eye aspect ratio technique," in *2020 IEEE Student Conference on Research and Development (SCOReD)*, 2020, pp. 448–452.

[53] A. Kashevnik, I. Lashkov, A. Ponomarev, N. Teslya, and A. Gurtov, "Cloud-based driver monitoring system using a smartphone," *IEEE Sensors Journal*, vol. 20, no. 12, pp. 6701–6715, 2020.

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

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

[56] K. Kim, T. Baltrusaitis, A. Zadeh, L.-P. Morency, and G. Medioni, "Holistically constrained local model: Going beyond frontal poses for facial landmark detection," University of Southern California, Institute for Robotics and Intelligent . . . , Tech. Rep., 2016.

[57] G. J. Edwards, C. J. Taylor, and T. F. Cootes, "Interpreting face images using active appearance models," in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, 1998, pp. 300–305.

[58] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[59] J. Alabort-i Medina and S. Zafeiriou, "Bayesian active appearance models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3438–3445.

[60] G. Tzimiropoulos and M. Pantic, "Gauss-newton deformable part models for face alignment in-the-wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1851–1858.

[61] Y. Wu and Q. Ji, "Facial landmark detection: A literature survey," *International Journal of Computer Vision*, vol. 127, no. 2, pp. 115–142, 2019.

[62] M. Wimmer, S. Fujie, F. Stulp, T. Kobayashi, and B. Radig, "An asm fitting method based on machine learning that provides a robust parameter initialization for aam fitting," in *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*, 2008, pp. 1–6.

[63] P. Tresadern, P. Sauer, and T. Cootes, “Additive update predictors in active appearance models,” in *Proceedings of the British Machine Vision Conference.* BMVA Press, 2010, pp. 91.1–91.12, doi:10.5244/C.24.91.

[64] D. Cristinacce and T. F. Cootes, “Boosted regression active shape models.” in *BMVC*, vol. 2. Citeseer, 2007, pp. 880–889.

[65] M. Dantone, J. Gall, G. Fanelli, and L. van Gool, “Real-time facial feature detection using conditional regression forests,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* Providence, RI, USA: IEEE, 2012, pp. 2578–2585.

[66] V. Kazemi and J. Sullivan, “One millisecond face alignment with an ensemble of regression trees,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1867–1874.

[67] J. M. Saragih, S. Lucey, and J. F. Cohn, “Deformable model fitting by regularized landmark mean-shift,” *International journal of computer vision*, vol. 91, no. 2, pp. 200–215, 2011.

[68] D. Cristinacce, T. F. Cootes *et al.*, “Feature detection and tracking with constrained local models.” in *Bmvc*, vol. 1, no. 2. Citeseer, 2006, p. 3.

[69] Y.-L. Tian, T. Kanade, and J. F. Cohn, “Facial expression analysis,” in *Handbook of face recognition.* Springer, 2005, pp. 247–275.

[70] W.-C. Lai, “Integrated charging with neural networks for fusion sensors of driver monitoring system,” in *2021 IEEE 12th Energy Conversion Congress & Exposition-Asia (ECCE-Asia).* IEEE, 2021, pp. 2065–2068.

[71] E. E. Galarza, F. D. Egas, F. M. Silva, P. M. Velasco, and E. D. Galarza, “Real time driver drowsiness detection based on driver’s face image behavior using a system of human computer interaction implemented in a smartphone,” in *International Conference on Information Technology & Systems.* Springer, 2018, pp. 563–572.

[72] S. Mehta, S. Dadhich, S. Gumber, and A. Jadhav Bhatt, “Real-time driver drowsiness detection system using eye aspect ratio and eye closure ratio,” in *Proceedings of international conference on sustainable computing in science, technology and management (SUSCOM), Amity University Rajasthan, Jaipur-India*, 2019.

[73] T. Issenhuth, V. Srivastav, A. Gangi, and N. Padoy, “Face detection in the operating room: Comparison of state-of-the-art methods and a self-supervised approach,” *International journal of computer assisted radiology and surgery*, vol. 14, no. 6, pp. 1049–1058, 2019.

[74] T. Sri Mounika, P. Phanindra, N. Sai Charan, Y. Kranthi Kumar Reddy, and S. Govindu, "Driver drowsiness detection using eye aspect ratio (ear), mouth aspect ratio (mar), and driver distraction using head pose estimation," in *ICT Systems and Sustainability.* Springer, 2022, pp. 619–627.

[75] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 687–10 698.

[76] H. Bai, M. Cao, P. Huang, and J. Shan, "Self-supervised semi-supervised learning for data labeling and quality evaluation," *arXiv preprint arXiv:2111.10932*, 2021.

[77] "Fairness: Type of bias," https://developers.google.com/machine-learning/crash-course/fairness/types-of-bias, accessed: 2022-06-18.

[78] S. Pan and D. Madill, "Generalized sliding window algorithm with applications to frame synchronization," in *Proceedings of MILCOM '96 IEEE Military Communications Conference*, vol. 3, 1996, pp. 796–800 vol.3.

[79] M. F. S. Sabir, I. Mehmood, W. A. Alsaggaf, E. F. Khairullah, S. Alhuraiji, A. S. Alghamdi, A. El-Latif *et al.*, "An automated real-time face mask detection system using transfer learning with faster-rcnn in the era of the covid-19 pandemic," *Computers, Materials and Continua*, pp. 4151–4166, 2022.

[80] J. Pan, L. Xia, Q. Wu, Y. Guo, Y. Chen, and X. Tian, "Automatic strawberry leaf scorch severity estimation via faster r-cnn and few-shot learning," *Ecological Informatics*, vol. 70, p. 101706, 2022.

[81] H.-H. Lin, T. Zhang, Y.-C. Wang, C.-T. Yang, L.-J. Lo, C.-H. Liao, and S.-K. Kuang, "A system for quantifying facial symmetry from 3d contour maps based on transfer learning and fast r-cnn," *The Journal of Supercomputing*, pp. 1–21, 2022.

[82] X. Xu, M. Zhao, P. Shi, R. Ren, X. He, X. Wei, and H. Yang, "Crack detection and comparison study based on faster r-cnn and mask r-cnn," *Sensors*, vol. 22, no. 3, p. 1215, 2022.

[83] L. Xiong, M. Ye, D. Zhang, Y. Gan, and Y. Liu, "Source data-free domain adaptation for a faster r-cnn," *Pattern Recognition*, vol. 124, p. 108436, 2022.

[84] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision.* Springer, 2014, pp. 740–755.

[85] E. Eidinger, R. Enbar, and T. Hassner, "Age and gender estimation of unfiltered faces," *IEEE Transactions on information forensics and security*, vol. 9, no. 12, pp. 2170–2179, 2014.

[86] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen *et al.*, “Iarpa janus benchmark-b face dataset,” in *proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 90–98.

[87] “Microsoft face api,” . https : / / www.microsoft.com / cognitive - services / en-us/faceapi.

[88] “Watson visual recognition,” https://www.ibm.com/watson/products-services.

[89] “Face++ api,” http://old.faceplusplus.com/ demo-detect/.

[90] T. B. Fitzpatrick, “The validity and practicality of sun-reactive skin types i through vi,” *Archives of dermatology*, vol. 124, no. 6, pp. 869–871, 1988.

[91] K. Kärkkäinen and J. Joo, “Fairface: Face attribute dataset for balanced race, gender, and age,” *arXiv preprint arXiv:1908.04913*, 2019.

[92] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, “Yfcc100m: The new data in multimedia research,” *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.