Analyzing Racial Bias in Facial Recognition Algorithms

Scott Morris

December 2021

1 Introduction

Over the past decade, the use of artificial intelligence (AI) and machine learning in our everyday lives has skyrocketed from when it first began to be studied. We are now capable of creating algorithms that surpass human performance at certain tasks and have added them to many of the objects we use such as phones, cars, and watches. None of these advancements could have been possible without feeding these models massive amounts of data about us. This includes our browsing activity, how we write and text, and even the pictures we upload online. However, these algorithms can have negative consequences when the data it's trained on is biased towards specific populations or excludes others. My project focuses on the racial bias and ethical issues that exist in facial recognition algorithms currently being used by law enforcement, government, and large tech companies. Motivation on this topic arises from the publishing of articles and research papers authored by scholars, researchers, and software engineers on how these algorithms are a passive extension of the institutional and societal biases that exist. To learn more about how facial recognition algorithms can show bias, I attempt to create my own balanced facial recognition model using racial classification and convolutional neural networks.

2 Background

2.1 Facial Recognition

Facial recognition is a technology that falls under computer vision, detecting and recognizing human faces from images or live camera feed. There are two main types of facial recognition, one-to-one matching and one-to-many. One-to-one matching involves analyzing a face to see if it's a match to a specific person. This is commonly used today for security features such as our phones or door locks. One-to-many involves looking for a best match of a person's face against a large dataset of faces and is primarily used by law-enforcement to find crime suspects. While facial recognition certainly has its positive uses, it has come

under fire for racial profiling and invasion of privacy especially when used by law enforcement.

2.2 Racial Bias in Facial Recognition

In recent years, facial recognition is increasingly being used by law enforcement, and without little question as to whether the algorithm is actually correct. In January of 2020 Robert Williams, a black man living in Michigan was falsely arrested due to a facial recognition algorithm matching him to an image of a suspect. Despite his adamant refusal and knowledge that he was not who they were looking for, he was still taken into custody because "the algorithm says it's you". This incident led to ACLU filing a lawsuit against the police department claiming that the arrest broke Williams' Fourth Amendment and civil rights [1]. Williams is just one of three cases in the state alone where facial recognition has led to a false arrest and in each case so far, it has happened to black men.

However, it's no hidden secret to these institutions that their algorithms are skewed. In 2019, the National Institute of Standards and Technology (NIST) reviewed all facial recognition systems with findings that the majority performed poorly on matching non-white faces [2] and performed the worst on black women. When you take into account that 1 in every 2 American adults are in a law enforcement facial network, [3] one can only imagine the effects wide-scale use of these facial recognition algorithms can have on populations that are already subjected to profiling bias by law enforcement.

Another concerning issue about the use of facial recognition is that there's currently no federal legislation as to how it can be used and the datasets that train it are created. This means that the photos we post online are free to be taken and used to feed these models without our knowledge or consent. It allows companies such as Clearview.AI, which claims to have the world's largest facial network with over 10 billion faces gears their product towards law enforcement [4]. As artificial intelligence and machine learning continue to be applied to new areas and slowly becomes more integrated into society, we must recognize and address the ways it inherently reflects systems of racial and societal bias.

2.3 Technical Overview

2.3.1 Deep Learning

The topics discussed in this paper relate to deep learning and image classification. Deep learning is a machine learning technique that gained popularity for its continuous performance increase with the more data gained compared to

[1] Ryan-Mosley, Tate. "The New Lawsuit That Shows Facial Recognition Is Officially a Civil Rights Issue." MIT Technology Review, MIT Technology

[2] Hao, Karen. "A US Government Study Confirms Most Face Recognition Systems Are Racist." MIT Technology Review, MIT Technology Review, 2 Apr. 2020

[3] "The Perpetual Line-Up." Perpetual Line Up, Georgetown Law, https://www.perpetuallineup.org/

[4] "Clearview AI — The World's Largest Facial Network." Clearview AI.

previous approaches where performance would rise and then stay at the same level. This allows us to gain more insight into the features present in the data. Deep learning is used in a wide range of technologies such as natural language processing and computer vision.

2.3.2 Artificial Neural Networks

Artificial Neural networks are a type of deep learning algorithm modeled after the human brain. Like how our brain uses neurons to process information, artificial neural networks use nodes to process data, and these nodes are linked to other nodes and so on. Every neural network has three main parts: An input layer, hidden layers, and an output layer. The input layer is the data that the neural network will be processing. The hidden layer is where weights are calculated onto nodes based on the information gained by the data and the use of various mathematical functions that can be applied depending on the desired goal. The output layer is the culmination of the weights that come before it to provide some sort of information about the data from the input layer. Neural networks can be applied to a large range of tasks and have even been optimized over time depending on the task, including facial recognition.

2.3.3 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a type of neural network most commonly used for analyzing images and image classification due to their ability to detect and make sense of patterns. This is because of the convolutional layers, hidden layers within the model. As can be seen in Figure 1, a convolutional layer gains data from images by using a grid filter to only look at a few pixels of an image at a time. This filter takes the dot product of the weights and stores that value then slides to the right and performs the same operation on the next set of pixels until it has a full matrix of dot products as its output. Max pooling layers work hand in hand with convolutional layers by reducing the dimensionality of images by decreasing the number of pixels from the previous convolutional layer. Similar to convolutional layers, max pooling layers also use a grid filter to look at the dot products outputted by the convolutional layer and takes the highest value present and stores it before moving to the next set.

3 Literature Review

While machine learning algorithms are constantly being tested and improved upon, there are only a handful of papers and articles that address the racial and societal biases reflected through widely used commercial algorithms.

3.1 Pilot Parliaments Benchmark

Some of the work being done to raise awareness on the lack of fair data representation in popular machine learning algorithms include Gender Shades. Created

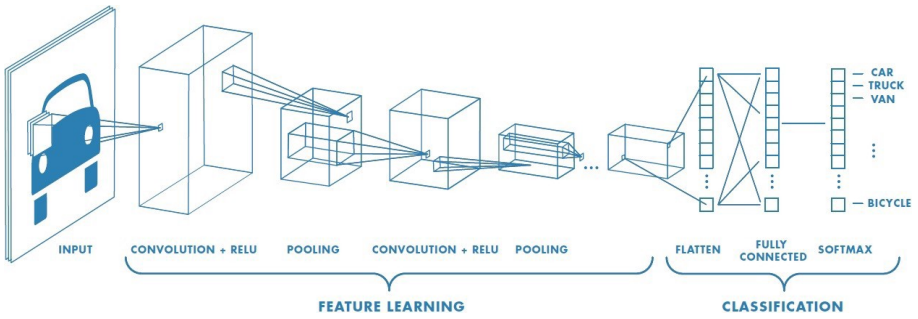


Figure 1: Convolutional Neural Network Architecture

by Joy Buolamwini and Timnit Gebru, AI researchers and founders of the Algorithmic Justice League, the project shows racial and gender bias present in the facial recognition algorithms published by Microsoft, Google, and IBM as well as evaluate the diversity of skin colors present in large facial datasets that exist. They used the Fitzpatrick Skin Type (FST) to evaluate the content of two large facial dataset benchmarks, IJB-A and Adience and found that for both datasets, 80 percent of the images were of lighter-skinned faces [5]. To combat and correct the disparities seen in popular facial networks, they created the Pilot Parliaments Benchmark (PPB), an image dataset which contains 1,270 unique faces taken from the official pictures of Parliament members of African and European countries and classified them by gender and FST. Their results found that the commercial algorithms tested performed much worse for darker-skinned women than the near perfect accuracy rates for lighter-skinned men. These results show the need to improve the quality of datasets and more research as to why these disparities exist before being used for commercial use.

3.2 Fair Face

Other work being done to correct algorithmic bias is Fair Face, which sets out to create a more diverse facial dataset balanced on race, age, and gender. The dataset contains around 108,000 faces taken from public image datasets such as Yahoo's *YFCC100M* and images posted on Twitter by media outlets. The images were then split into seven racial classifications: White, Black, East Asian, Hispanic/Latino, Middle Eastern, and Southeast Asian. Unlike many of the large facial datasets that exist publicly, their dataset is the only one to be balanced as well as have non-merged racial classifications. Compared to these other large facial recognition algorithms, it performed better classification accuracy on all aspects which suggests that having more diverse datasets can

[5] Buolamwini, Joy, and Timnit Gebru. "Gender Shades: Intersectional accuracy disparities in commercial gender classification." Conference on fairness, accountability and transparency. PMLR, 2018.

improve performance[6]. This suggests that the to improve overall performance, we should focus on making datasets more inclusive and balanced across features.

4 Methods

4.1 Libraries

I created my model using Python and used a combination of standard and non-standard libraries. The non-standard libraries I imported for use were Tensorflow, Keras, Matplotlib, Sklearn and Numpy. Tensorflow and Keras were used for the creation of my model as these libraries allow for the application of deep learning techniques and neural network creation. I used Sklearn, Numpy, and Matplotlib to allow for visualization of my model's results and accuracy through tables.

4.2 Data Creation and Pre-processing

When planning out the architecture of my model, I first began with where by image data will come from and how representational it was in terms of content. My model will be performing one-to-many matching, as it will be comparing faces based on the image data and racial classifications it's seen before as an output so it was important to have enough diverse data. Many of the public datasets that exist online did not have much representations in terms of skin color present in the images or the dataset was dominated by images of specific groups. I also had to take into account that many of the photos in these datasets are *in the wild*, meaning that they vary greatly in terms of quality, setting, lighting, and size. For these reasons I decided my best option was to use the Fair Face dataset to train my model due to its image diversity being much higher others. Another upside of using this dataset is that all the images are already classified by race, are the same size (244 x 244), and already split into train and validation folders. I then began to create my own training, validation, and test sets using a small subset of the whole data. Within these sets, I created seven new folders; each titled with the race labels present in the dataset and placed images into their corresponding folder. I chose to use race as my classification label over FST labeling because it would give better insight into how specific racial groups are affected by bias and because skin color can vary based on image quality.

My first step was to pre-process my images through Keras to confirm that my test and validation set images and labels are correctly found and recognized. For my first set of tests, the training set contained 1,400 images, the validation set contained 280 images, and the test set 140. The number of images for each classification was equally balanced as well throughout the folders. I declared the batch size of my model as 200, which declares how many images the model trains

[6]Kärkkäinen, Kimmo, and Jungseock Joo. "FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age." ArXiv.org, 14 Aug. 2019, arxiv.org/abs/1908.04913

itself on at a time. Generally, the larger the batch size the better performance at the cost of computing time.

4.3 Convolutional Neural Network Model Creation

To create my facial recognition model, I used Tensorflow and Keras, open-source libraries that allow for the application of deep learning technology and creation of neural networks in Python. I then began to define the layers and their parameters for my convolutional neural network. I decided to make my model Sequential, meaning that each layer has one input and output tensor which would allow. The first hidden layer in my model is a 2D Convolutional Layer. The arguments I defined for this layer were the number of filters, kernel size, the activation and padding. It has 32 filters, meaning it has 32 outputs to the next layer, and a kernel size of 3x3 which defines the size of the convolutional window that will go over the pixel data of the input images. The choice to use 32 filters based on previous tests and found it to perform better than using 16 or 64 in both processing time and accuracy. I made the kernel size 3x3 as that is commonly used to process image data in convolutional neural networks. The activation function for the layer is 'relu' and padding equal to 'same', which means the images have no padding and allows for the dimensionality of the images to not be reduced by the convolution operation. Following our convolutional layer is its complimentary max-pooling layer which has a defined pooling size of 2 x 2, of which it will take the highest value found per pixel group. The layer is also defined to have 2 strides which means it moves 2 spaces across the image which prevents overlap between pixel values that have already been included in the max-pooling. For the second convolutional layer, the only difference is doubling the number of filters to 64 as it is common practice to increase the number as you go into deeper into the hidden layers of the model, and paired it with an identical max-pooling layer. I then flattened the data which turns the pooled feature map of values into a one-dimensional column. The final output layer of the model is a dense layer, meaning it's fully connected to all the nodes of the previous layer. The output layer uses soft-max activation function to evaluate the probabilities of each possible output and has 7 nodes, one for each racial classification. My model is also

4.4 Evaluation

Following model initialization, I then specified how its performance will be evaluated when compiled. For my loss function, I chose categorical cross-entropy due to having multiple classes rather than binary. It works similar to the soft-max function by calculating the probabilities of each class across all possible options. The metric I used for evaluation was accuracy due to having my model predict use racial classification. To help prevent my model from over-fitting during training, I added an early callback method which would automatically stop training once it began to see signs of over-fitting. I set my model to have a default of 25 epochs, the number of forward and backward passes through the

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 224, 224, 32)      896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 112, 112, 32)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 112, 112, 64)      18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 56, 56, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 200704)            0
_________________________________________________________________
dense (Dense)                (None, 7)                 1404935
=================================================================
Total params: 1,424,327
Trainable params: 1,424,327
Non-trainable params: 0
_________________________________________________________________
```

Figure 2: Final model architecture with layers and total number of parameters

training and validation sets provided, but it averaged around 11 to 12 epochs to stop over-fitting using the callback method. From my tests fitting the model on training and validation data, I could see that while the loss and accuracy on the training set improved greatly, the validation set struggled much more as can be seen in the figure below.

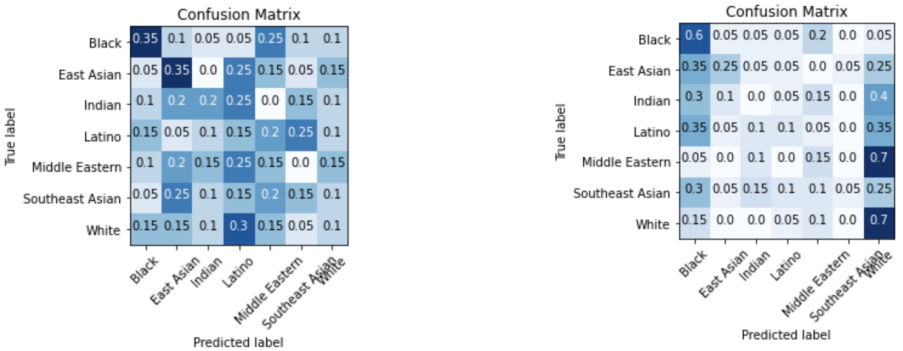
```
model.fit(x=train_batches, validation_data=valid_batches, epochs = 25, verbose=2, callbacks=[earlystopping])
Epoch 1/25
7/7 - 137s - loss: 569.5353 - accuracy: 0.1300 - val_loss: 218.3476 - val_accuracy: 0.1393
Epoch 2/25
7/7 - 139s - loss: 69.5594 - accuracy: 0.1579 - val_loss: 6.0322 - val_accuracy: 0.1143
Epoch 3/25
7/7 - 111s - loss: 3.3535 - accuracy: 0.1807 - val_loss: 1.9962 - val_accuracy: 0.1214
Epoch 4/25
7/7 - 108s - loss: 1.9459 - accuracy: 0.1657 - val_loss: 1.9545 - val_accuracy: 0.1750
Epoch 5/25
7/7 - 136s - loss: 1.9433 - accuracy: 0.1614 - val_loss: 1.9502 - val_accuracy: 0.1679
Epoch 6/25
7/7 - 154s - loss: 1.9425 - accuracy: 0.1571 - val_loss: 1.9479 - val_accuracy: 0.1464
Epoch 7/25
7/7 - 135s - loss: 1.9415 - accuracy: 0.1621 - val_loss: 1.9480 - val_accuracy: 0.1429
Epoch 8/25
7/7 - 97s - loss: 1.9373 - accuracy: 0.1593 - val_loss: 1.9507 - val_accuracy: 0.1536
Epoch 9/25
7/7 - 100s - loss: 1.9136 - accuracy: 0.2164 - val_loss: 1.9786 - val_accuracy: 0.1107
Epoch 10/25
7/7 - 90s - loss: 1.8098 - accuracy: 0.2957 - val_loss: 2.1498 - val_accuracy: 0.1143
Epoch 11/25
7/7 - 117s - loss: 1.5546 - accuracy: 0.4264 - val_loss: 2.2444 - val_accuracy: 0.1357
<keras.callbacks.History at 0x7f86c5d05d60>
```

Figure 3: Model loss and accuracy on training and validation sets

5 Results

To visualize the model's performance, I created a confusion matrix using Sklearn, Numpy, and Matplotlib which can be seen in Figure 4. I believe that this is one of the best ways to visualize the model's accuracy as it shows how the model predicted with the test set against each classification. The results shown come from using a small separate test set containing 140 images. The first set of tests

on the model in Figure 4a with a training set size of 1,400 spread equal across classifications, the model performed poorly overall, but higher accuracy with certain groups with the best being 35% accuracy with Black and East Asian faces. It performed the worst accurately matching White faces with 10% accuracy and conflating them as White. In Figure 4b, on a much larger training set of 7,000 images spread equally across images, the model still performed poorly, but found greater accuracy rates for certain classifications and even worse results for some. It correctly guessed 60% of black faces and 70% of white faces as white. However, it incorrectly predicted all Indian faces, 95% of Southeast Asian, 90% of Latino. Interestingly we can also see that a large percentage of the other race classifications were predicted as either White or Black by the model. Comparing the two tests we can observe that the smaller training set had less accuracy for specific classifications and more spread out predictions while the larger training set had much more perceived bias towards Black and White classifications. Reasons for poor overall performance of the model may be that the size of the training set size is not large enough to accurately classify the seven labels well enough. Another possibility is that the model is still be over-fitting the data, making it incorrectly conflate one classification's features with another. This shows how facial recognition models can still discriminate against populations even when using a dataset with equal representation across race. In America, law enforcement facial recognition algorithms are largely biased against people of color due to the datasets used for these algorithms having much more images containing white and lighter-skinned faces and not enough representation to accurately discriminate black and brown skin tones.



(a) Training Set Size: 1,400

(b) Training Set Size: 7,000

Figure 4: Test Set Confusion Matrices

6 Conclusion

It can be seen from the results that my attempts to make a racially balanced facial recognition algorithm were unsuccessful. Throughout the process of cre-

ating my model, I ran into a few limitations. One was GPU speed as formatting the original dataset to fit Keras' data pre-processing format made it impossible to use the whole dataset and on average, it took my laptop close to 20 minutes to complete training on the small training set and an hour on the larger. This also relates to the processing and energy power required for deep learning algorithms with large amounts of data. I also found it difficult at times to assess what changes to make to my model such as batch sizes or number of filters affected performance as I saw little overall improvements. A consideration I had for my model was to not use the early callback method to force my model to run the specified amount of epochs, complete passes through the training data, at the risk of over-fitting. If I hadn't used my early callback method, the model was set to train for 25 epochs. Out of curiosity I ran my model allowing it to run all 25 epochs on the large training set and saw that validation set accuracy ended at 21%. The results were not especially better from what we saw in Figure 4, resembling 4a in terms of accuracy and prediction distribution across classifications. I believe that if I was able to use the entire Fair Face dataset, I would have observed better results on the test set due to my training set having more data to base its predictions for each classification.

In conclusion, facial recognition is currently in a stage of growth. As seen from previous studies, racial bias is very prevalent in commercial facial recognition and can create negative experiences for everyday people. Many people argue about the ethics of its use and whether it should even be allowed. With the lack of representation in many datasets being used by law enforcement, it is pivotal that these algorithms are reassessed to not be so biased towards underserved populations. In fact, large tech companies such as IBM, Amazon, and Microsoft have stopped selling their facial recognition algorithms to algorithms to law enforcement or taken a pause from the research altogether [7]. While there is still yet to be any federal legislation as to what facial recognition can be used for, the FDA released "guiding principles" for machine learning practice. Some notable inclusions are the use of multi-disciplinary expertise, separate training and testing sets, and that the model takes into account the impacted user population [8]. This is a step in the right direction if we want to make technologies that are equitable for everyone.

References

[1] Ryan-Mosley, Tate. "The New Lawsuit That Shows Facial Recognition Is Officially a Civil Rights Issue." MIT Technology Review, MIT Technology

[2] Hao, Karen. "The New Lawsuit That Shows Facial Recognition Is Officially a Civil Rights Issue." MIT Technology Review,

[7] Hill, Kashmir, and Ryan Mac. "Facebook, Citing Societal Concerns, Plans to Shut Down Facial Recognition System." The New York Times, 2 Nov. 2021. NYTimes.com.

[8] Health, Center for Devices and Radiological. "Good Machine Learning Practice for Medical Device Development: Guiding Principles." FDA, FDA, Oct. 2021. www.fda.gov,

https://www.technologyreview.com/2021/04/14/1022676/robert-williams-facial-recognition-lawsuit-aclu-detroit-police/.

[3] "The Perpetual Line-Up." Perpetual Line Up, https://www.perpetuallineup.org/.

[4] "Clearview AI — The World's Largest Facial Network." Clearview AI, https://www.clearview.ai.

[5] Buolamwini, Joy, and Timnit Gebru. "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification." Proceedings of the 1st Conference on Fairness, Accountability and Transparency, PMLR, 2018, pp. 77–91. proceedings.mlr.press, https://proceedings.mlr.press/v81/buolamwini18a.html.

[6] Karkkainen, Kimmo, and Jungseock Joo. "FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age for Bias Measurement and Mitigation." 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2021, pp. 1547–57. DOI.org (Crossref), https://doi.org/10.1109/WACV48630.2021.00159.

[7] Hill, Kashmir, and Ryan Mac. "Facebook, Citing Societal Concerns, Plans to Shut Down Facial Recognition System." The New York Times, 2 Nov. 2021. NYTimes.com, https://www.nytimes.com/2021/11/02/technology/facebook-facial-recognition.html.

[8] Health, Center for Devices and Radiological. "Good Machine Learning Practice for Medical Device Development: Guiding Principles." FDA, FDA, Oct. 2021. www.fda.gov, https://www.fda.gov/medical-devices/software-medical-device-samd/good-machine-learning-practice-medical-device-development-guiding-principles.