APS360 - Final Report Part A: FaceRace - Group 2

Nicholas Ishankov 1005798521, Amir Seken 1005893561, Kareem Elmaaddawy 1005728321, Osman Afandiyev 1004507786

Word Count: 2405

1 Introduction

Mobile devices and social media services having biometric recognition has become an expectation and is nothing new, but 10 years since facial recognition was first used by companies like Facebook, it is still the least accurate biometric parameter [1]. Traditionally facial scans are used to categorize users based on gender and race in policing for law enforcement, but we could see it employed for marketing and product development with companies like IBM, Amazon, and Facebook developing facial recognition algorithms [2]. Demographic and ethnic information can be determined using facial recognition and used in targeted advertising making it a relevant problem to solve. Also, predicting ethnicity for ethnically-diverse countries like Canada is relevant as there is a lack of ethnic data in existing databases [3]. Current models lead to racial discrimination and target incorrect audiences, as we have seen already in American law enforcement [2], and so our group employed machine learning to determine the most likely ethnic matches given a facial scan. Ethnicity classification is a problem that requires a lot of data, which not all applications will have access to, and the problem deals with sensitive data to make decisions which makes it hard to implement a heuristic model thus machine learning is a feasible tool (Figure 1) [3]. Current machine learning models are suspected to be trained with unbalanced data and we plan to mitigate this with multiple datasets and balancing data for all races and genders.

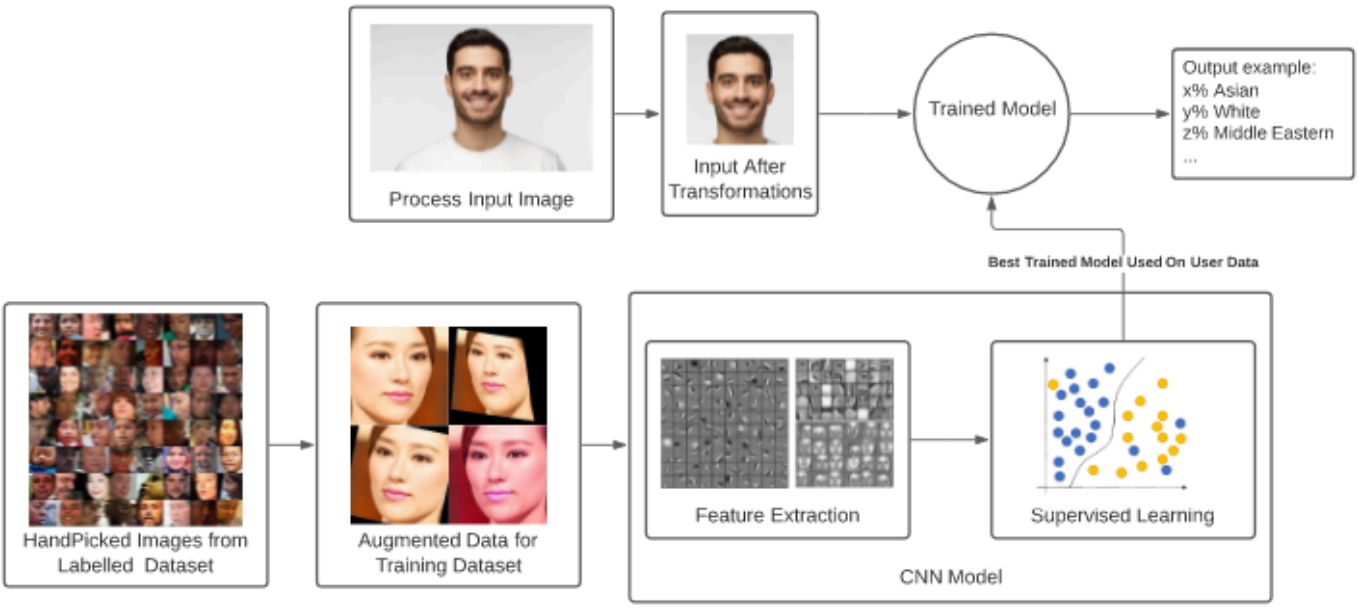

Figure 1: Project idea diagram and example output given input image [4-7].

2 Background & Related Work

Facial recognition algorithms are inaccurate due to the fact that these algorithms fall victim to the 'cross race effect' - where the performance of an algorithm is better on people closely related to the geographical area of where it was created [8]. To mitigate this, we have made sure to incorporate balanced datasets. The importance of such incorporation is supported by two studies [8,9] that have worked on the

same scope as our project. In addition to this, both studies have come to the consensus that the images fed into the model must be pre-processed in a certain way which mainly involved cropping out irrelevant backgrounds, centering of the face and in some cases, heuristically determining the area around the face (hair color, texture and structure). Furthermore, both models involved using a state of the art convolutional neural network architecture (SNET) which is widely known to be preferred for computer vision applications [10]. Although we initially planned to emulate such an architecture, we ended up resorting to an architecture similar to that of VGG11.

3.0 Data Processing

To assemble our dataset we collected images of peoples faces of distinct races from multiple datasets. We made sure that the total number of samples for each of the races is the same, to minimize bias [11]. We divided our dataset into training, validation and test sets. Due to many images being unusable (Figure 3), the team handpicked images with forward facing faces and little obstructions and then performed data augmentation to increase the training set size. The training set had 400 (2000 after augmentation) images per class and the validation and test set had 220 per class making the total images used 12200.

3.1 Datasets

The following are the datasets that we chose to preprocess and were selected as they have the most balanced split between races (Figure 2).

1. UTKFace Dataset [12]

    - The dataset consists of over 20,000 images with annotations of age, gender, and ethnicity. This dataset has images that cover large variations in pose, facial expression, etc.
    - We will clean off this dataset from the age and gender labels, since we only identify races. Then we will include its samples into our dataset.

2. FairFace Dataset [13]

    - A face image dataset which is race balanced. It contains 108,501 images from 7 different race groups: White, Black, Indian, East Asian, Southeast Asian, Middle Eastern, and Latino.
    - Images were collected and labeled with race, gender, and age groups.
    - We will clean off this dataset from the age and gender labels, since we only identify races. And add the filtered samples into our dataset.

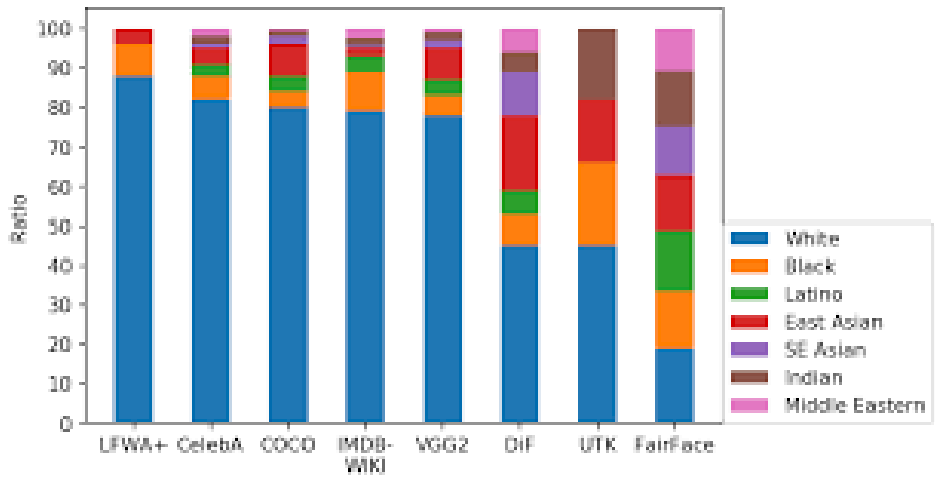

Figure 2: Ratio of different races in different Datasets (right most are UTK and FairFace)[14].

We handpicked images with less obstructions and more visible facial features (eyes, mouth, nose, etc.) to be able to teach our model these key features using less resources (Figure 3). These images were transformed to 224x224 in the data loader.

Figure 3: Usable image (left) compared to unusable image (right) from our dataset.

3.2 Data Augmentation

To increase the amount of data for the training set, we have performed 4 types of image augmentations for each hand picked image in our training dataset. Figure 4 illustrates an original image in our dataset and 4 augmentations performed on it.

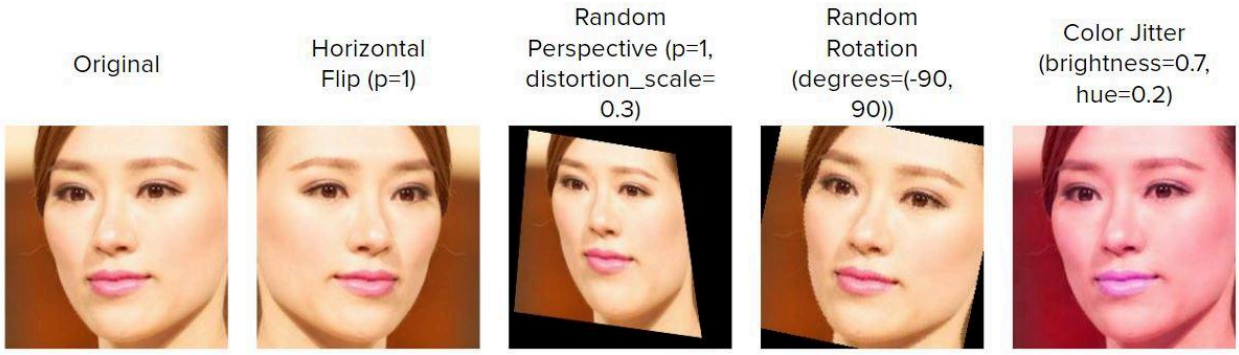

Figure 4: Example augmentations along with exact torchvision.transforms parameters.

The first three augmentations were intended to help the model to learn facial features specific to each race while the last augmentation has been made to prevent skin color being the only learned feature for the predictions and increase the model's capability to generalize.

4.0 Model Architectures

Common solutions to image classification problems involve implementing your own convolutional neural network (CNN) architecture, transfer learning with a pretrained model, or fine-tuning an existing model. We decided to build our own CNN model, and as an alternative, implement a pre-trained transfer learning model.

4.1 Transfer Learning (VGG16)

We used a transfer learning approach to see if we will get higher accuracy compared to the CNN model. We've chosen a pretrained VGG16 model as a feature extractor and added a simple ANN classifier (Figure 5). It is easy to implement and it achieved one of the highest accuracies on ImageNet competition [15]. The ANN configuration and VGG architecture are in Figures 6 and 7.

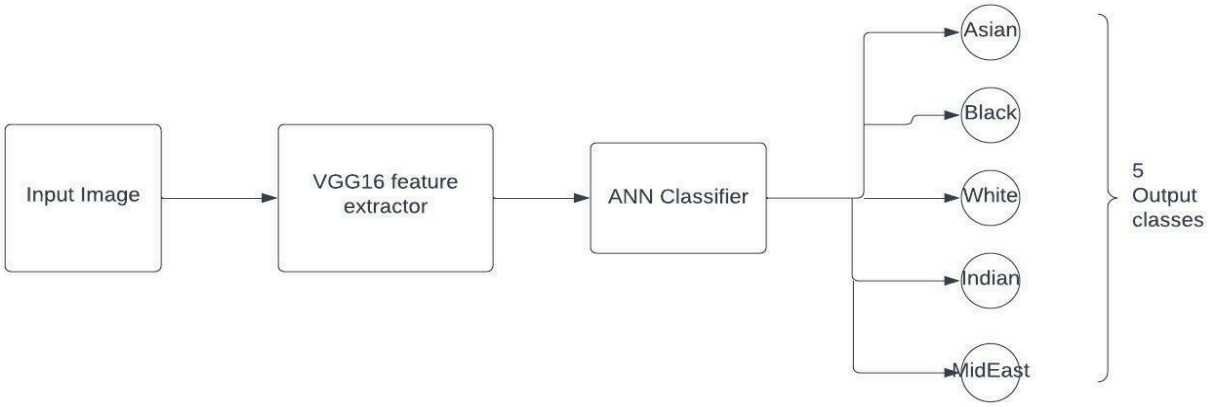

Figure 5: High-level architecture of transfer learning model.

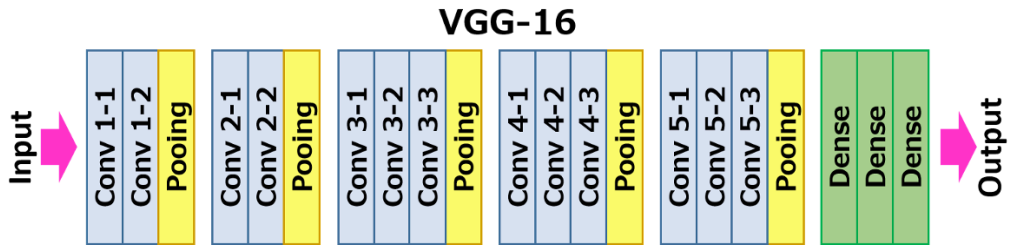

Figure 6: Inner architecture of VGG16 feature extractor [16].

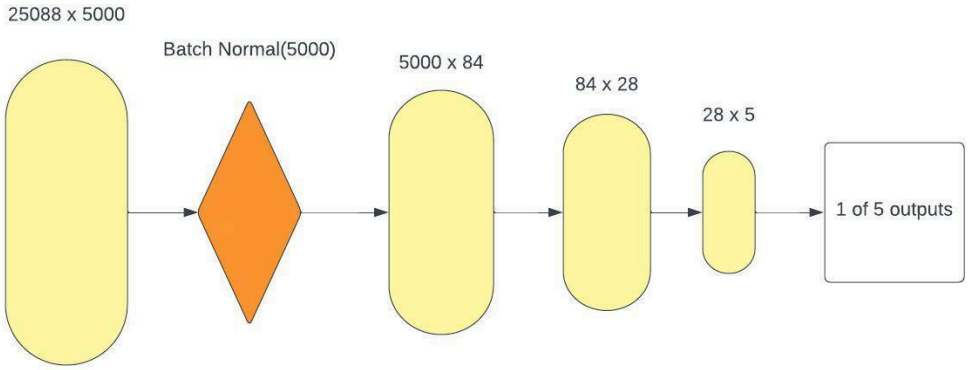

Figure 7: Architecture of ANN Classifier.

4.2 Convolutional Neural Network

Our model takes as input our cropped 224x224 images and classifies the subjects as one of the five classes mentioned earlier. The first eight layers are convolutional, which learn filters across spatial and channel dimensions, while the remaining three are fully-connected linear layers, which learn to classify the images based on the learned visual features, as shown in Figure 8. As seen from the green rectangles in the figure, we decided to incorporate batch normalization which helped greatly in avoiding distortion in the input distribution between layers. Additionally, to lower the dimensionality and extract the most important features from the convolutions we added max pooling. On the fully-connected layer side, nodes at the input and hidden layers were dropped with a probability of 25% to prevent them from synchronously optimizing their weights.

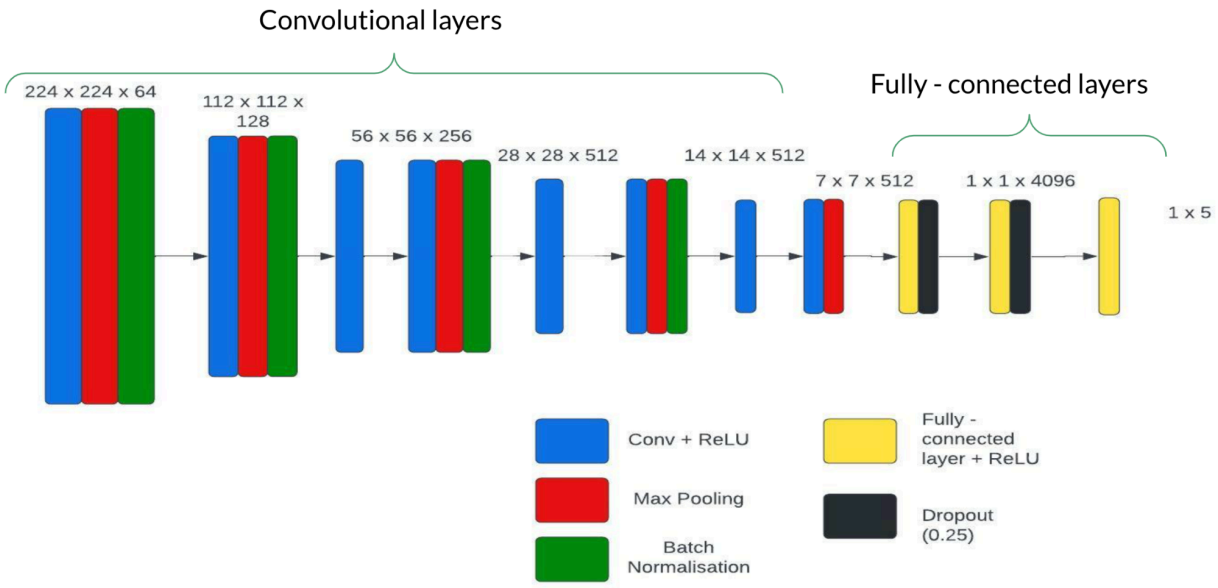

Figure 8: CNN architecture followed by fully-connected layers

5 Baseline Model

The team developed two baseline models initially. The first model was an ANN which would process grayscale images. The ANN model was performing better, 44%, on the validation dataset than randomly picking, 20%, from all possible races. However, as CNNs are typically utilized in computer vision problems due to their ability to extract features with filters, and ability to prevent the loss of valuable information by grayscaling the input images, we decided to switch to a shallow CNN as our baseline model. The new baseline had just three convolutional layers with max-pooling after each convolutional layer and two fully connected layers. Maximum pooling layers were used to down-sample information between each convolutional layer to improve efficiency. The relu activation function was used to be able to learn non-linear transformations. The best configuration of hyperparameters (lr = 0.001, batch size = 64, epochs = 15) revealed the validation accuracy of 53% for the baseline model.

6.0 Quantitative Results

The team had created a transfer learning model, and a CNN model to perform race classification. The results for each of our models show that the CNN model is able to attain better validation accuracy, so the CNN model was used for evaluating the model on our test dataset. The accuracy represents a model's ability to correctly guess the race associated with the face out of 5 possible races.

6.1 Transfer Learning (VGG16)

The transfer learning model was able to attain a training accuracy of 99.4% and a validation accuracy of 65% after training the model for 5 epochs.

Looking at Figure 9, the validation accuracy saturates at 65% at 5 epochs and the training curve reaches 99% accuracy. Increasing the number of epochs to 10 showed the curve stay here indicating we were training our model to its maximum potential with our existing data. We tried experimenting with different hyperparameters (dropout in ANN, data normalization), but the highest validation accuracy we managed to get is 65%.

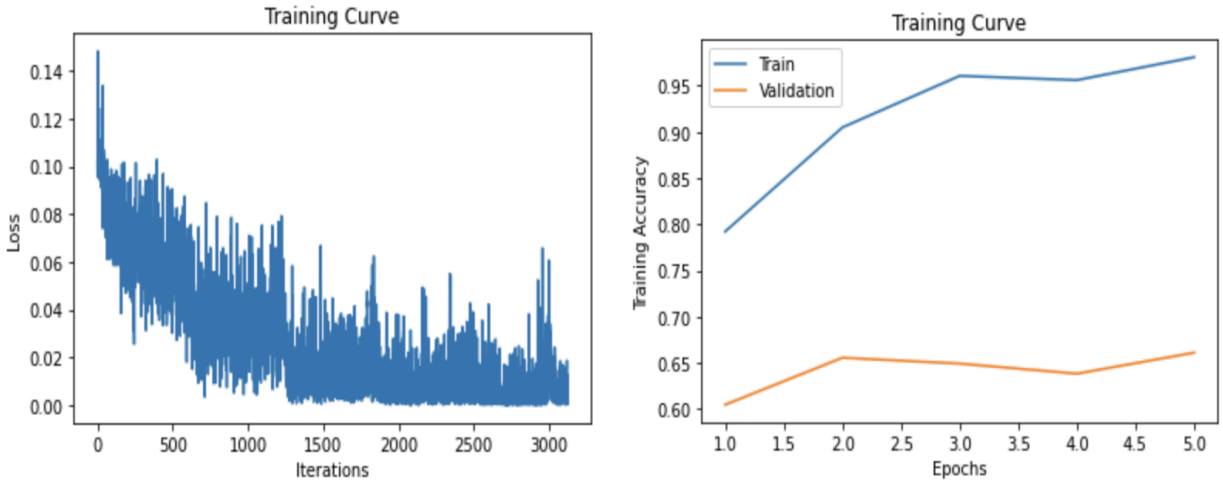

Figure 9: Loss and training accuracy curves of transfer learning model.

6.2 Convolutional Neural Network

The convolutional neural network model was able to outperform our transfer learning implementation with training and validation accuracies of 99.9% and 75.7% respectively (Figure 10). These numbers were a result of constant tweaking of hyperparameters such as the epoch number, learning rate, dropout rates, and normalization layers. Final values we settled on were 20, 0.005 and 0.25 respectively. This model was saved and later evaluated on our test dataset yielding a test accuracy of 71.8%.

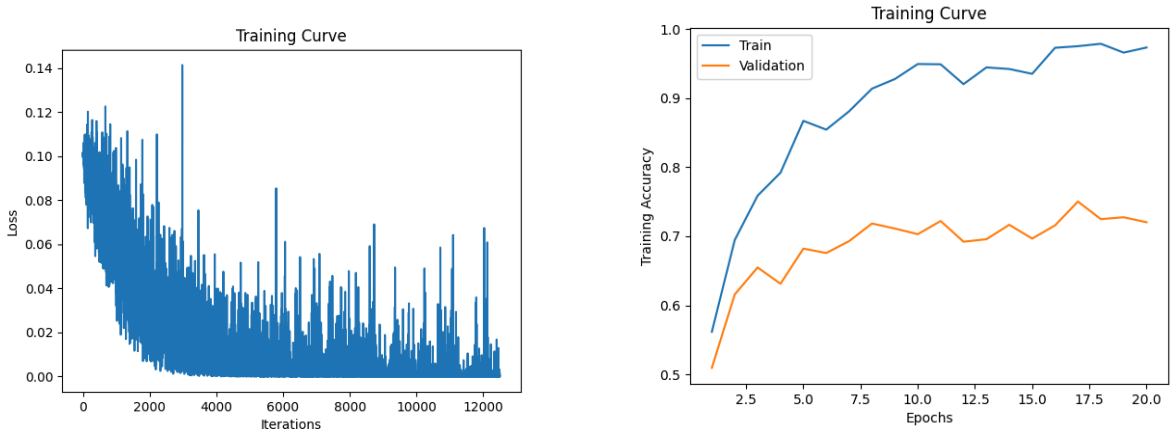

Figure 10: Loss, training and validation accuracy curves of CNN model

7 Qualitative Results

To understand how well our model is performing we created a function to determine a confusion matrix using the models' test accuracies for each class. Initially, we were working with 6 different classes and the model was able to get a test accuracy of around 58% while the validation accuracy was 68%. The model curves were saturated indicating that the model was being pushed to its limit and this difference in accuracy suggests the model being biased. This was already the best hyperparameter configuration (dropout, data normalization, batch normalization layers) so we were considering that there could be an issue with the data. The confusion matrix for 6 classes (Figure 11) suggests that the model lacks data to be able to differentiate between the Latino class, and Indian class. By dropping the Latino class and using the same hyperparameters, but reducing the number of outputs, the model's test accuracy improved to 72% and its validation accuracy was 76%. We were able to use more data for each class which eliminated bias and improved performance. In Figure 12 we can see that this change improved the model accuracy for every class.

Asian	**55.83**	4.16	8.33	9.16	3.33	19.16
Black	0.0	**82.5**	5.0	6.66	2.5	3.33
Latino	4.16	15.0	**38.33**	30.0	11.66	0.83
Indian	6.66	10.0	6.66	**47.5**	25.83	3.33
Middle Eastern	2.5	2.5	9.16	20.0	**60.83**	5.0
White	0.83	0.83	5.83	9.16	19.16	**64.16**
	Asian	Black	Latino	Indian	Middle Eastern	White

Figure 11: Confusion matrix with the model performance for 6 classes for CNN Model.

Asian	**79.09**	9.55	1.36	6.36	3.64
Black	2.74	**84.47**	5.94	5.48	1.37
Indian	6.85	9.13	**58.90**	14.16	10.96
Middle Eastern	7.27	1.82	11.36	**70.0**	9.55
White	7.72	3.64	6.82	15.0	**66.8**
	Asian	Black	Indian	Middle Eastern	White

Figure 12: Confusion matrix with the model performance for 5 classes for CNN Model.

8 Demonstration

The team has created two mechanisms to evaluate the performance of the model on the dataset. The first one is presenting the model's correct prediction accuracies per class and over the entire dataset. The second mechanism displays each image in the dataset in a grid format with the model's prediction and true race of that image as a header. The second mechanism was intended to enable the tester to visualize the model's performance on individual images and generate insights on the model's overall performance.

To ensure a good representation of the model's performance two key goals were set by the team before creating the "demonstration" dataset. The first goal was to ensure that no demonstration dataset image was present in either validation or training set. This would enable us to prevent the model from being tested on the data that it was trained with. If this precaution was not taken, such a demonstration dataset could have resulted in higher accuracy and given the illusion of a better model, while the actual reason for the high accuracy could have simply been the model's familiarity and according weight adjustments for those images. The second goal was to ensure that the images picked for the demonstration dataset matched the standards of the image used for training and validation (Figure 3). With these two core ideas in mind, the team searched the web and picked the images of famous people and added the images of team members and their relatives. Each such image was examined with the previously mentioned two requirements prior to being added to the demonstration dataset. As a result, the team obtained the demonstration dataset of 30 images – 6 per class. The total accuracy on the demonstration dataset of the final model was 80% (Figure 14). Accuracy per class is presented in the table below:

Asian	Black	Indian	MiddleEastern	White
100%	100%	83.33%	50%	66.67%

Figure 13: Accuracy per class for demonstration dataset.

Total accuracy on the demonstration dataset exceeded our expectations and beat the test accuracy, 72%, of the final model. This pattern of behaviour can be explained due to the small size of the demonstration dataset, 30 images, which might have resulted in accidentally picking the type of face images that the model was better tuned to predict.

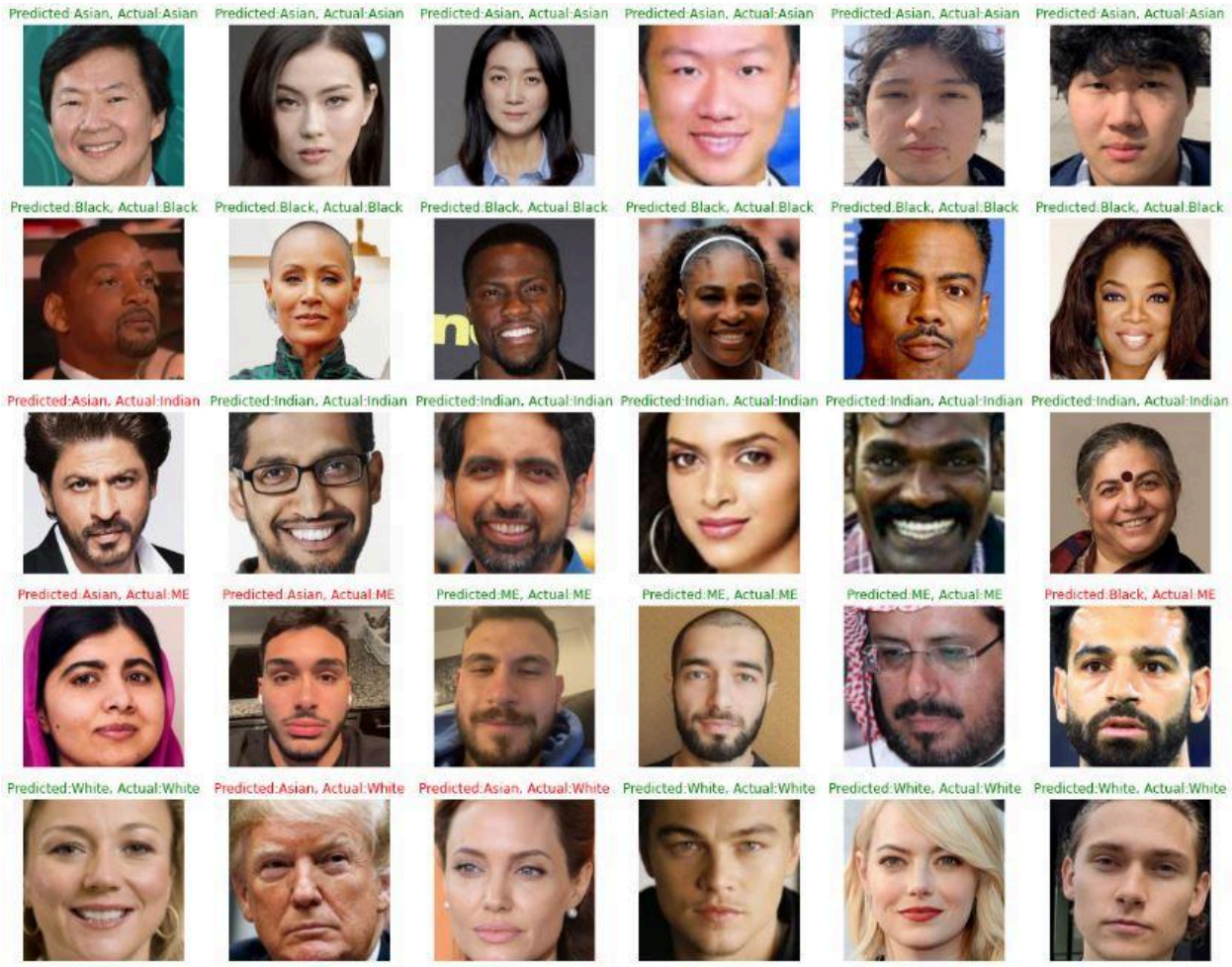

Figure 14: Grid view of all images in demonstration dataset.

9 Discussion

Handpicking a smaller set of more usable images, then growing the dataset with data augmentation, and using our own CNN had yielded the best results. The transfer learning model that we implemented attained a 65% validation accuracy compared to the CNN's 76% validation accuracy. It underperforms since VGG16 is trained on millions of images and many aren't related to this task so it struggles to generalize with our reduced dataset. An approach to better utilize transfer learning would be to fine-tune the VGG16 model for race estimation [9].

Our final CNN model was able to get a test set accuracy of 72% compared to an accuracy of 48% from the baseline. The model performed the worst on Indian and White faces, and best on Asian and Black (Figure 12). The problem is trickier than we expected and to achieve these results we had to abandon using the entire dataset of approximately 60,000 images and reduced the number of classes from 6 to 5. With reduced classes, we were able to increase the number of images per class while still taking the same amount of time to train our model. Although, due to bad images in the original dataset (Figure 3), handpicking images greatly reduced the total number of images per class – 400, and, hence, the accuracy of the model, augmenting the dataset, thereby increasing the number of images per class – 2000, in train dataset, proved to be effective and increased the validation accuracy from 67% to 76% for the final model.

A study with a model classifying 4 races was able to achieve 97% accuracy after training on a dataset of 1.3 million images [9]. Having more time and computational resources, in order to process more data, would yield better results for a deep CNN like ours. Furthermore, given the sensitivity of this problem and using the study mentioned as a benchmark, it would be highly unethical to deploy our model to the outside world.

10 Ethical Considerations

Putting the terms “race/ethnicity” and “machine learning” in the same sentence would naturally raise a lot of red flags. This is why we have made sure to be ethically conscious with our implementation as using such a model could give rise to ethical issues. For example, if the training datasets exclude a race, then the tools using the model could empower disaggregation which in turn contributes to racist narratives. Additionally, any research involving people must go through the process of individual informed consent [17], therefore the use of anonymized secondary data can hinder such a process.

11 References

[1] “A brief history of facial recognition - NEC New Zealand,” *NEC*, 14-May-2021. [Online]. Available: https://www.nec.co.nz/market-leadership/publications-media/a-brief-history-of-facial-recognition/. [Accessed: 03-Feb-2022].

[2] A. says: R. P. says: cjenk415 says: P. G. says: R. K. says: R. H. says: A. G. says: N. P. says: A. khattak says: A. Says: T. shyara says: and V. L. says: “Racial discrimination in face recognition technology,” *Science in the News*, 26-Oct-2020. [Online]. Available: https://sitn.hms.harvard.edu/flash/2020/racial-discrimination-in-face-recognition-technology/. [Accessed: 08-Feb-2022].

[3] K. O. Wong, O. R. Zaïane, F. G. Davis, and Y. Yasui, “A machine learning approach to predict ethnicity using personal name and census location in Canada,” *PLOS ONE*. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0241239. [Accessed: 08-Feb-2022].

[4] “Fairface: Face attribute dataset for balanced race, gender ...” [Online]. Available: https://openaccess.thecvf.com/content/WACV2021/papers/Karkkainen_FairFace_Face_Attribute_Dataset_for_Balanced_Race_Gender_and_Age_WACV_2021_paper.pdf. [Accessed: 05-Feb-2022].

[5] “How Convolutional Neural Networks work,” *KDnuggets*. [Online]. Available: https://www.kdnuggets.com/2016/08/brohrer-convolutional-neural-networks-explanation.html/2. [Accessed: 05-Feb-2022].

[6] L. & Justin, “Machine learning for beginners: Overview of algorithm types,” *Just into Data*, 26-Nov-2021. [Online]. Available: https://www.justintodata.com/machine-learning-algorithm-types-for-beginners-overview/. [Accessed: 05-Feb-2022].

[7] “Face man images, stock photos & vectors,” *Shutterstock*. [Online]. Available: https://www.shutterstock.com/search/face+man. [Accessed: 05-Feb-2022].

[8] A. S. Darabant, D. Borza, and R. Danescu, “Recognizing human races through machine learning-a multi-network, multi-features study,” *MDPI*, 19-Jan-2021. [Online]. Available: https://www.mdpi.com/2227-7390/9/2/195. [Accessed: 09-Feb-2022].

[9] M. A. Ahmed, R. D. Choudhury, and K. Kashyap, “Race estimation with Deep Networks,” *Journal of King Saud University - Computer and Information Sciences*, 24-Nov-2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157820305589. [Accessed: 09-Feb-2022].

[10] “Deep Learning for Computer Vision,” *Run.ai*. [Online]. Available: https://www.run.ai/guides/deep-learning-for-computer-vision#:~:text=Convolutional%20Neural%20Networks%3A%20The%20Foundation,to%20traditional%20image%20processing%20algorithms. [Accessed: 09-Feb-2022].

[11] Index. (n.d.). Retrieved February 8, 2022, from http://whdeng.cn/RFW/index.html

[12] Utkface. (n.d.). Retrieved February 8, 2022, from https://susanqq.github.io/UTKFace/

[13] Papers with code - fairface dataset. Dataset | Papers With Code. (n.d.). Retrieved February 8, 2022, from https://paperswithcode.com/dataset/fairface#:~:text=FairFace%20is%20a%20face%20image,%2C%20gender%2C%20and%20age%20groups

[14] “Fairface: Face attribute dataset for balanced race, gender ...” [Online]. Available: https://openaccess.thecvf.com/content/WACV2021/papers/Karkkainen_FairFace_Face_Attribute_Dataset_for_Balanced_Race_Gender_and_Age_WACV_2021_paper.pdf. [Accessed: 05-Feb-2022].

[15] R. Thakur, “Step by step VGG16 implementation in Keras for beginners,” *Medium*, 24-Nov-2020. [Online]. Available: https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c?source=list-85370dce2b14----a833c686ae6c----3-------71f6283852e0----------. [Accessed: 12-Apr-2022].

[16] “Transfer learning using VGG16 in Pytorch.” [Online]. Available: https://zephyrnet.com/vi/chuy%E1%BB%83n-giao-vi%E1%BB%87c-h%E1%BB%8Dc-b%E1%BA%B1ng-vgg16-trong-pytorch/. [Accessed: 12-Apr-2022].

[17] J.-C. Mariátegui, “Cybernetics and systems art in Latin America: The Art and Communication Center (CAYC) and its pioneering art and Technology Network - AI & Society,” *SpringerLink*, 01-Feb-2022. [Online]. Available: https://link.springer.com/article/10.1007%2Fs00146-021-01341-7. [Accessed: 09-Feb-2022].