# Face-Classifier Report

Mohamad Ghanmeh    Nigel Yong Sao Young

## Abstract

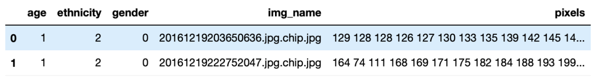
*Machine Learning (ML) is a growing field that has reached many with the recent advancement in technology and is able to give accurate predictions to the unimaginable. Our project's main goal is to leverage ML techniques and algorithms to develop an optimal age, gender and ethnicity Face Classifier. Some of the techniques include Data Augmentation and Transfer Learning.*

*We use both conventional ML models and Deep Learning models, namely, Decision Trees, Random Forests, Logistic Regression, Support Vector Machines and Deep Convolutional Neural Networks. Finally, the best results we obtained was approximately **90%** accuracy for the Gender, **79%** accuracy for the Ethnicity and **6** years mean absolute error (MAE) for the Age predictions.*

## 1. Introduction

In this report we aim to provide a clear understanding of how we built an age, gender and ethnicity face classifier. We thoroughly describe the "what" and the "why" of the steps we took to reach our goal.

The data we are working with is a cleaned version of the Face UTK dataset [5] found on Kaggle [2]. It is ready to be used and does not need data cleaning. Here is a preview of the content of the CSV file in Figure 1.



| | age | ethnicity | gender | img_name | pixels |
|---|---|---|---|---|---|
| **0** | 1 | 2 | 0 | 20161219203650636.jpg.chip.jpg | 129 128 128 126 127 130 133 135 139 142 145 14... |
| **1** | 1 | 2 | 0 | 20161219222752047.jpg.chip.jpg | 164 74 111 168 169 171 175 182 184 188 193 199... |

Figure 1: Dataset preview

- age column contains values ranging from 1 to 116 years old classification task
- ethnicity column contains integers from 0 to 4, representing White, Black, Asian, Indian and Others (like Hispanic, Latino, Middle Eastern, etc) respectively.
- gender column contains integers 0 or 1, representing Male and Female respectively.
- img_name column contains the image name
- pixels represents a list of 2304 pixels (48 * 48) in total, representing the grayscale image.

Here is a figure showing the types of images provided to have an idea of the pictures the dataset holds.

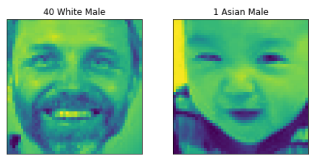


Figure 2: Sample images with labels

In figure 2, the images has their corresponding age, ethnicity and gender of the individuals in the title. It is exactly what we aim to predict from someone's face image.

Our goal is to experiment with particular ML techniques to maximize the prediction performance. In the next section of the report, we will discuss thoroughly about our methods, experiments and results.

## 2. Methodology

In this section, we are going to elaborate on the different steps we used to reach our goals which helped us in maximizing the prediction performance.

### 2.1. Data Analysis

To start working with the data, it's important that we first understand the data. This means knowing about the characteristics that the dataset might hold which would later on help us to maximize prediction performance.

We first got to know how the dataset looked like as explained in the introduction, and how many samples the dataset had and the number of pixels each image had. Then we plotted the distributions of each features with histograms.

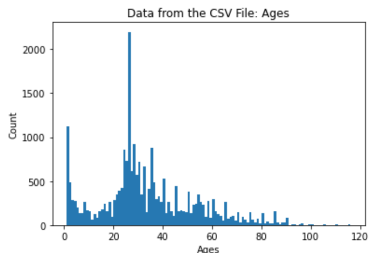


Figure 3: Age distribution

Figure 3 depicts that the dataset is mostly formed with 26 and 1 years old individuals. We noted that this might result in biased predictions.

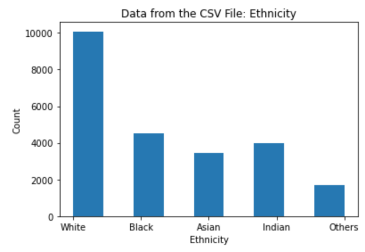


Figure 4: Ethnicity distribution

Above, we observed that most people in the dataset are mostly white, while we have less of the 'others' category.

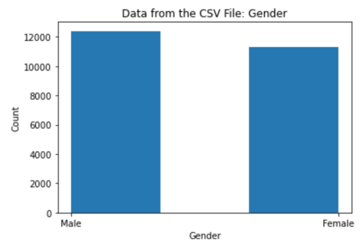


Figure 5: Gender distribution

In figure 5, on the other hand, the distribution seems to have a good balance between Male and Female, with Male being slightly higher but shouldn't produce too much of an impact on the results.

## 2.2. Data Splitting and Pre-processing

Secondly, before implementing any ML techniques we split and preprocess the image data. To do so, we arbitrarily split the data into 80% training and 20% testing. Both the training and testing would have X and y. X being the image pixels itself and y being the target values: Age, Gender and Ethnicity.

Next, to preprocess the images, we standardize features by removing the mean and scaling to unit variance. This step is critical as some models assume that all features are centered around 0 and have variance in the same order such as the RBF kernel of Support Vector Machines. [4]

## 2.3. Training with scikit-learn models

Next, we actually train with multiple scikit-learn models. Namely, Decision Trees (DT), Random Forests (RF), Logistic Regression (LR) and Support Vector Machines (SVM). We created a general function that would train the model given and run a grid search cross validation on the given parameters. The function helps us determine the best set of attributes which gives the best results. It also prints the best validation accuracies of each best model, which helps us compare the models.

## 2.4. Training with CNNs

For image classification, Convolutional Neural Networks (CNN) are the pretty much the state-of-the-art nowadays. We use TensorFlow [1] by Google for building and training the CNNs. We created a general function, build_model, for all 3 predictions which creates the same CNN but with different loss functions, activation functions and number of classes given as parameters. To be precise:

- The gender model uses sigmoid as activation function, binary cross entropy as loss function since it is a binary classification task
- The age model uses ReLU as activation function and mean square error as loss function since it is a linear regression task
- The ethnicity model uses softmax as activation function and sparse categorical cross entropy as loss function since it is a multi-class classification task

After much experimentations, our final CNN architecture is inspired from the VGG neural network, which is has one of the best benchmarks in image classification. Deeper you get in the network, the more are the number of filters and the smaller are the filters dimensions. At the end of the model, we then have fully connected layers.

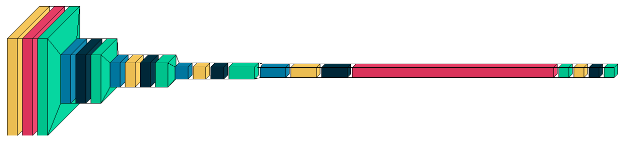


Figure 6: Our CNN Macro Architecture

A more detailed diagram can be found in the Appendix. 13 We also augment the input (as discussed in 3.1.2) before the first convolutional layer and have dropouts and batch normalization after pooling.

## 2.5. Transfer Learning

Since we had little amount of data, we decided to use Transfer Learning, from which we use a saved network that was previously trained on a large dataset. The first layers in a neural network is mostly very generalized features, we make use of the 'bottom' layers of the neural network and use the 'top' layers for our classification purpose.

After training the model with the bottom layers being "freezed", it was able to reach only 70% accuracy at best unfortunately. To further improve it, we then unfreeze some of the middle layers to train them and 'fine-tune' the model. However, the accuracy worsened and got stuck at around 51%.

We tried multiple pre-trained models by Tensor-Flow such as MobileNetV2 and ResNet50 but none of them gave better results than what we previously had with CNNs.

## 3. Experiments

In this section, we will discuss about the 'variables' of the project. Basically, how we tried to experiment with some techniques that would potentially yield us with better accuracies in prediction.

### 3.1. Splitting data among groups

From section 2.1, we analyzed the data and observed how imbalanced it was regarding the ethnicity distributions. Specifically, the White category had almost 5 times more samples than the Others category: around 10000 samples as opposed to around 2000 samples. Refer to figure 4. To solve this imbalance, we used two techniques.

#### 3.1.1 Stratifying

Instead of only randomly splitting the training and testing with a 80/20 ratio, we added the stratify attribute on the ethnic groups. Put simply, this would result in having a proportional random amount of Ethnic groups in both the training and testing split. While it did not have any very noticeable results to our prediction accuracies, it was logical for us to use it.

#### 3.1.2 Undersampling

Second, we experimented with undersampling the White ethnic group before splitting. This method is really a trade-off between having less data and having a less biased data. It proved to have noticeable results in evaluation of the models. When looking at the confusion matrix, it immediately made the predictions less bias to predict White ethnicity.

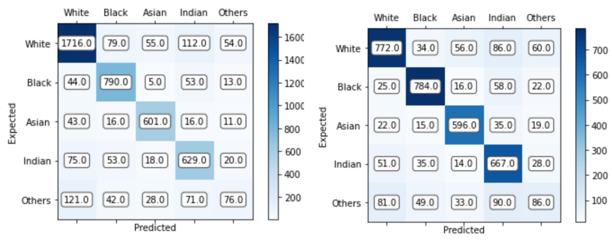


Figure 7: Without (Left) & With (Right)

We notice darker colours in the diagonals with undersampling.

### 3.2. Data augmentation

In general, since we relatively have low amount of data (only 20k images), we had to augment the data and also compensate for the samples lost in undersampling as discussed in the section above. Many data augmentation techniques were available: rotation, change in saturation, brightness, cropping and flipping — among others. The ones which was most appropriate for us was flipping horizontally and rotating slightly. This technique did not have a tremendous change to the prediction accuracies but made sense to keep.

## 4. Evaluation

It was imperative to compare the different models. Validation and testing accuracy is definitely a good indicator but we're also concerned with other evaluation metrics in our case. As a whole, here are the accuracies for each models. (For Age it is set to be the mean absolute error (MAE))

| Models | Age (MAE) | Ethnicity % | Gender % |
|---|---|---|---|
| DT | 14.4 | 53.3 | 73 |
| RF | 12.7 | 65.7 | 81.6 |
| LR | 10.5 | 72.3 | 85.1 |
| SVM | 11.5 | 74.3 | 86.7 |
| CNN | 6 | 79 | 89 |

Table 1: Model Accuracy Comparison table.

At this point, we concluded that our best models were CNNs and SVMs.

### 4.1. Gender Evaluation

To further evaluate the best gender models, we did two things: plot their confusion matrix and classification report.

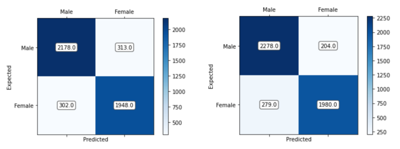


Figure 8: SVM (Left) & CNN (Right) Confusion Matrix

As we can see, the CNN made slightly less errors than the SVM. Both models were not specifically biased towards one gender or another.

```
              precision    recall  f1-score   support

           0       0.88      0.87      0.88      2491
           1       0.86      0.87      0.86      2250

    accuracy                           0.87      4741
   macro avg       0.87      0.87      0.87      4741
weighted avg       0.87      0.87      0.87      4741
```

(a) SVM

```
              precision    recall  f1-score   support

           0       0.89      0.92      0.90      2482
           1       0.91      0.88      0.89      2259

    accuracy                           0.90      4741
   macro avg       0.90      0.90      0.90      4741
weighted avg       0.90      0.90      0.90      4741
```

(b) CNN

Figure 9: Classification report

Looking at the classification report, both of the models had a satisfactory precision, recall and f1-score. The CNN model turned out to be some numbers ahead of SVM model.

## 4.2. Age Evaluation

From the Mean Absolute Error, we already know that the CNN model is the best one. We can have a better evaluation of the models by plotting their predictions, to expected ages.

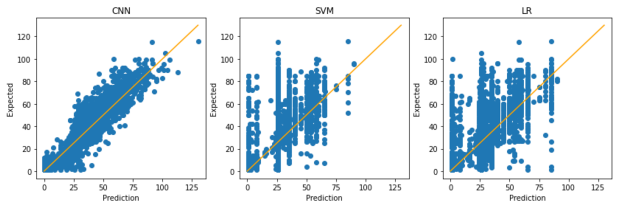


Figure 10: Age Evaluation

The only model which came close to the line, y=x, was the CNN. The SVM and LR model were very far off. The SVM and LR model were heavily biased with the age distribution, predicting most of the ages at 26 years old and almost none at 90+ years old.

## 4.3. Race Evaluation

Similar to the gender evaluation, we have a look at the confusion matrix and classification report.

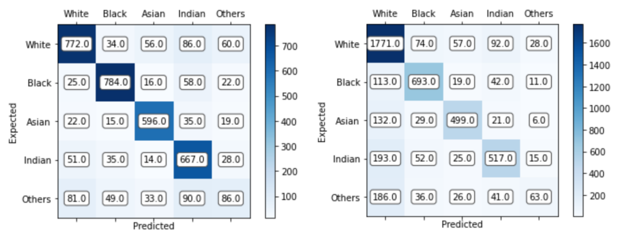


Figure 11: CNN (Left) & SVM (Right)

From the diagonal of the confusion matrix, we concluded that the CNN performed way better.

## 4.4. Error Analysis

The last step of evaluation was to visualize the errors that we predictions had and tried to see a pattern.

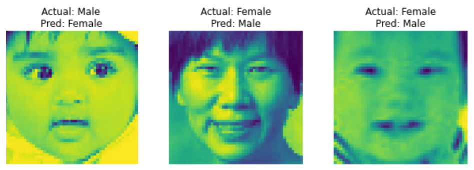


Figure 12: Visualizing Gender Errors.

One observation was that most of the errors are babies and are indeed pretty hard to identify.

## 5. Conclusions

Finally, we are quite satisfied with the accuracies achieved. That is, 90% accuracy for the gender, 6 years MAE for the age and 79% accuracy for the race. Secondly, we were also able to create an interactive way for a user to use the models. The user would be able to take a selfie from his webcam from the Jupyter itself. Lastly, we experimented with Transfer Learning and splitting the data among groups as we initially aimed. So we are quite content with it.

> We can use ML to make incredible predictions yet could not have predicted what difficulties we would have while working on the project beforehand. - Nigel Yong

If we were to start the project all over again, we would try to start with a better balanced dataset, with more training samples. We believe the dataset was certainly a limiting factor. Secondly, in Transfer Learning, we would start off a more specific pre-trained model such as FaceNet [3], which is trained only on Faces.

All in all, we are only able to talk about what we could or could not have done now because of all the knowledge acquired throughout the project. We surely learned a whole lot and are very happy to have reached this level of completion with the project. It has been an incredible introduction to ML for the two of us!

## References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, and et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 2

[2] Nipun Arora. Age, gender and ethnicity (face data) csv, Sep 2020. 1

[3] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015. 4

[4] sklearn. sklearn.preprocessing.standardscaler. 2

[5] susanqq. Utk dataset, 2018. 1

## 6. Appendix

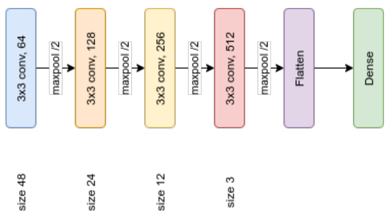


Figure 13: CNN Macro Architecture