NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

AY2021/2022 Semester 1

CE4042 - Neural Network and Deep Learning

Topic: Gender Classification Report

Done by:

Team Members	**Matriculation Number**
Mervyn Chiong Jia Rong	U1921023K
Yeo Li Ting	U1921466G
Elayne Tan Hui Shan	U1921730C

1. Introduction

Image-related problems require a large amount of time and resources for training models and fine-tuning parameters to achieve the required functionality to solve a given problem. However, in today's society, there are several existing CNN models that have been pre-trained to classify images. As such, we can utilise transfer learning techniques to employ the weights of pre-trained models and fine-tune them to our needs, thus saving computation time and overcoming sampling inadequacy.

2. Objective

Our project aims to solve a binary classification problem - to predict the gender of a human subject in an image, where the label is either 'male' or 'female'. Humans can easily distinguish between males and females in images; however, it is hard for us to pinpoint how we come to that conclusion. Additionally, relevant features are not always expressed in a standardized manner and the orientation and clarity of images differs from case to case. Therefore, we will be using the automatic gender classification, a deep learning algorithm that provides ways to process information and make accurate predictions without the need for predefined features.

We will be developing CNN models and testing them on CelebA and Adience datasets. CelebA has around 250,000 images of celebrities whereas Adience has around 26,000 images of around 2,000 subjects collected from the media hosting service, Flickr.

The main goals of this project are:

1. Build a robust CNN model from scratch
2. Determine the optimal parameters for the model
3. Apply hyperparameter tuning
4. Apply transfer learning on CelebA dataset
5. Test the model on Adience dataset
6. Evaluate the effects of transfer learning

3. Existing Techniques

3.1 Adience Benchmark

Gil Levi and Tal Hassner[1] did a detailed survey of varied gender and age classification techniques employed in the past to solve their classification problem. Some of the techniques employed include general machine learning techniques such as Support Vector Machines (SVM), AdaBoost, and Computer Vision approaches. These methods are used for training and validating datasets with high uniformity and low variance between images. The dataset images are taken in a highly controlled environment - near-frontal, well-aligned, high-quality, and with no occlusion and good lighting conditions. In contrast, the images in Adience and CelebA datasets are more naturalistic and include external disturbances such as extreme variations in head pose and poor lighting conditions. With this variance, these methods may not provide the most optimal results for these datasets.

Levi and Hassner proposed the OUI-Adience dataset as a new benchmark dataset for age and gender classification. The Adience dataset consists of images uploaded to Flickr from smart-phone devices. The images were uploaded without prior manual filtering; hence the viewing conditions of these images are highly unconstrained.

They also published a lean network architecture to accompany the Adience dataset and provide good results for the dataset despite its variances. Furthermore, the lean network was able to avoid overfitting, which is a communal problem for machine learning on small datasets.

---

[1] Gil Levi, & Tal Hassner. (n.d.). (thesis). *Age and Gender Classification using Convolutional Neural Networks*. Retrieved November 15, 2021, from https://www.cv-foundation.org/openaccess/content_cvpr_workshops_2015/W08/papers/Levi_Age_and_Gender_2015_CVPR_paper.pdf

After analysis, the development of our model shall leverage the Adience benchmark dataset, and the baseline model published by Levi and Hassner.

4. Method and Implementation

4.1 Data Cleaning and Preprocessing Steps

4.1.1 Dataset

Our experiment uses the Adience dataset for training, testing and validation. The dataset consists of two types of data, text files and images. The text files contain gender and age labels, and other metadata such as name and file path of the corresponding image. The data in the text files have been pre-split into 5 folds. The pre-split text files will also be used for cross validation to compare the model performance with other known models. Adience dataset is used as a premature benchmark data for gender and age classification to allow for ease of replication of experiment.

4.1.2 Data Cleaning

The text files consist of a small number of records with “NaN” or empty values that will be removed from our dataset during our cleaning process. The images corresponding to these text files do not have a standardized size. Therefore, the image data will be cleaned by resizing the images to a standardized size of 256x256.

After cleaning the dataset, we randomly split the cleaned data into 20-80 partitions where 20% of the data is used for testing and 80% of the data is used for training.

4.1.3 Final Train and Test Dataset Formulation

We combined all 5 folds of 80% of the data used for training to generate the final train set. Similarly, we combined all 5 folds of 20% of the data used for testing to generate the final test set. By sampling 20% of each fold to form the final test dataset, it benchmarks our model, and simultaneously allows us to evaluate the performance of the optimal model for real world application. By using the 20-80 partition, we ensure that the test dataset and train dataset are mutually exclusive, which prevents any data leak between both. This avoids falsified evaluation of performance on the optimal model.

4.1.4 Train and Validation Dataset Formulation for Cross Validation (CV)

We performed 5-fold cross validation by choosing one of the individual 5-fold 80%-partitioned dataset as the validation set. The remaining 4 individual 80%-partitioned dataset are combined to act as the train subset. All individual partitions are used once as the validation set. In total, we will have 5 validation sets and 5 train subsets.

4.2 Data Augmentation

Since the Adience dataset has a small number of images, we employed data augmentation to increase the number of images for deep learning, with the aim that more data renders higher performance. As such, we take an existing image and run it through augmentation techniques such as rotating it by 90 degrees, mirror imaging it and gray scaling it. Data augmentation is usually performed on small datasets to increase the data size. To investigate the validity of this statement, we employed data augmentation on the CelebA dataset, a relatively large dataset, in our experiment.

Fig 4.2 Data Augmentation Example

4.3 Model Architecture

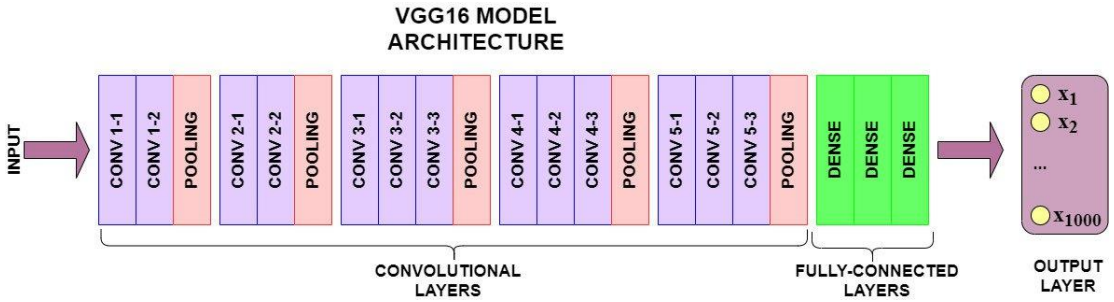

Fig 4.3.1 VGG Model Architecture

VGG16 architecture was used for our transfer learning on the CelebA dataset. It is a model that achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. To apply transfer learning, we froze all the layers of our model and were able to obtain an average accuracy of 90.8% for training data. This signifies a mere difference of 1.9% from what the reference model is able to achieve. Conversely, on average the validation accuracy is 71.93%.

4.3.1 Transfer Learning Application with Training and Validation

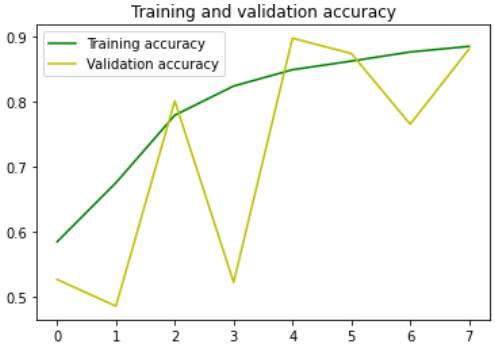

Fig 4.3.2 Training and Validation accuracy plot

With reference to Fig 4.3.2, the jagged validation accuracy clearly indicates overfitting when it peaks and then drops by a large margin. As our main goal was to simply observe the effects of transfer learning and how it will impact the Adience dataset, we did not tweak or change the hyper parameters in this portion of the experiment. The reason it stops at 8 epochs is due to the effect of early stopping which will interrupt the model fitting earlier than intended so as to prevent overfitting and underfitting.

4.3.2 Training and Validation Loss

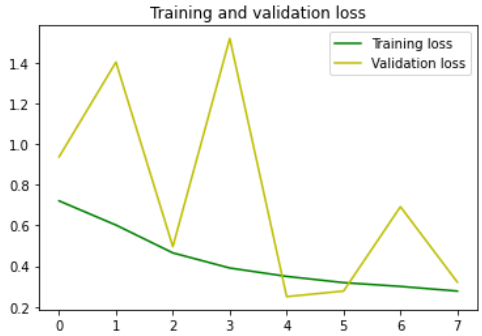

Fig 4.3.3 Training and Validation loss plot

Referencing Fig 4.3.3, training loss is steadily decreasing, which could be attributed to the marginal higher amount of data to work with in the training dataset that consists of 825,259 images, versus the validation dataset that consists of 87,653 images. Based on initial observations from epochs 1-3, validation loss is generally increasing despite a dip in epoch 2, and this suggests underfitting for the validation loss. However, after reaching epoch 8, it is evident that it is a case of overfitting as the graph tends towards a lower validation loss with occasional spikes.

4.4 Fine-Tuning

To round up transfer learning, we unfreeze the model layers and train the entire model with a low learning rate.

4.4.1 Test and Validation Accuracy

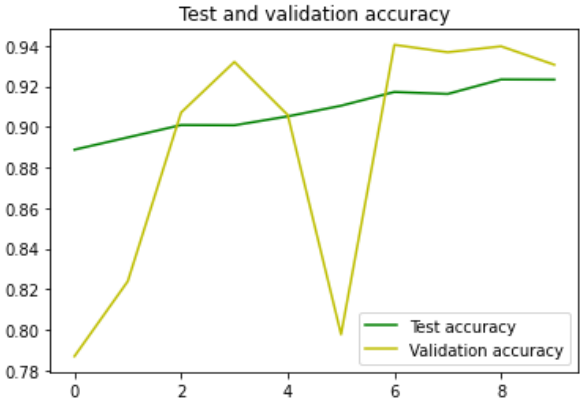

Fig 4.4.1 Test and Validation accuracy comparison

With reference to Fig 4.4.1, the test accuracy is similar to the training accuracy in Fig 4.3.2 as they both have a generally smooth gradient climb towards their saturation point and there are little variances and spikes in the data. The slight dip at epoch 7 hints at a potential of overfitting, but this possibility is ruled out as the test accuracy steadily approaches a saturation point of 92% and eventually achieves an average accuracy of 90.8%. One striking feature of this graph is the validation accuracy with peak accuracies of more than 92%, suggesting that it is tending towards a high validation accuracy. Overall, it achieved an average accuracy rate of 89%, and these results allow us to conclude that the fine-tuning was a success in improving our model accuracy.

4.4.2 Test and Validation Loss

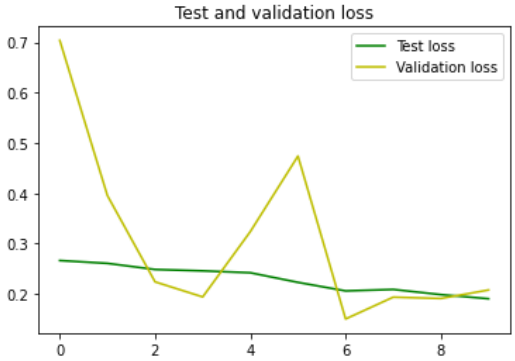

Fig 4.4.2 Test and Validation loss

Referencing Fig 4.4.2, when compared to figure 4.3.3, the validation loss performs significantly better as it has less spikes. It achieves an average lower loss rate of 30.56%. The spike in validation loss for epochs 4 and 5 suggests overfitting and the upward trajectory of validation loss from epoch 6 onwards proves that the validation loss is not ideal. Generally, the trend and gradient of the test loss is relatively smooth and gradually decreasing towards 0, thus it is safe to assume that no overfitting is occurring.
After conducting both experiments, we conclude that fine-tuning is a success but there is still room for improvement particularly in the case of validation data.

4.5 Model Selection

To determine the optimal hyperparameters for our model, we adjust our hyperparameters individually when conducting training on the train subsets to investigate its effect on our cross validation results. Early stopping is utilised to minimise validation error and prevent overfitting.
After analysing Levi's and Hassner's work, we implemented a baseline model based on the Lean CNN architecture mentioned in the research paper. We will then perform tuning of hyperparameters to modify the existing architecture, with the aim of improving the model's accuracy.

4.6 Tuning of hyperparameters

4.6.1 Tuning of Depth

In this portion, we experiment with the depth of our layers and investigate if removing Fully Connected (FC) layers and/or inserting more convolution layers and maxpool layers will aid in improving our model accuracy. As such, we have defined varying types of depth, namely 1fc, 2fc, 4conv, 1fc4conv, and 2fc4conv. 1fc and 2fc removes two and one FC layers respectively whereas 4conv adds one convolution layer and one maxpool layer before flattening occurs. 1fc4conv and 2fc4conv is a combination of 1fc and 4conv, and of 2fc and 4conv respectively. We found that 1fc4conv yields the best results.

4.6.2 Tuning of Width

For width, we experiment with the number of filters/neurons for our layers. In theory, for FC layers, the number of neurons determine the prediction capacity whereas for convolution layers, the number of filters affect the number of features learnt from the images. We investigate if this theory holds by tweaking these numbers. For comparison purposes, we half and double the original number of filters/neurons to clearly observe the effects caused to our model accuracy.

4.6.3 Tuning of Dropout

We have learnt that dropouts at FC layers are capable of enhancing our model performance. Therefore, we experiment with the dropout rates of the 2 original dropout layers, and add 1 extra dropout layer. We investigate the effects of dropouts by varying the dropout rates between (0, 0.2, 0.5, 0.8).

4.6.4 Tuning of Optimizer

For tuning of optimizers, we experiment with a few optimizers to study the effects of these optimizers on our model accuracy. Mini-batch Stochastic Gradient Descent (SGD) learning generally generates the best results, however it usually takes a lot of time for the model to converge. Hence, we experiment with variations, namely RMSProp, SGD, SGD with momentum, and Adam optimizer, on the Mini-batch SGD. We found that RMSProp yields the best results.

5. Experiment

Fold	Result	Overall Accuracy
0	0.84994	87.749%+-0.23%
1	0.8766	
2	0.8900	
3	0.8943	
4	0.8766	

5.1 Baseline Model

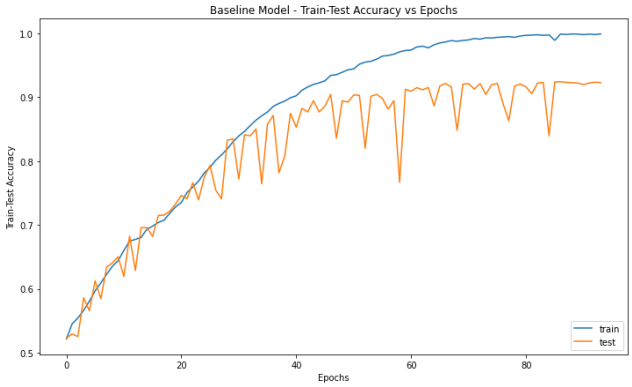

Fig 5.1.1 Baseline model Train vs Test Accuracy vs epochs

Referring to Fig 5.1.1, it is evident that the train accuracy managed to saturate towards 1.0 as compared to the test accuracy which is saturating but with multiple drops. It is observable that the test data suffers from overfitting, and this is also backed up by Fig 5.1.2 where a similar pattern is present in the loss graph plot. One thing to note is that early stopping for these graphs was set to 30 and as a result, did not activate to prevent overfitting from occurring.

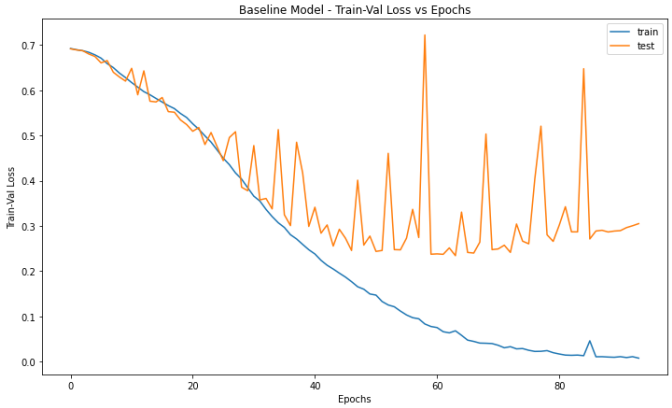

Fig 5.1.2 Baseline model Train vs Validation Loss vs epochs

5.2 Experiment on Depths

Depth Type	Model	Accuracy	Parameter count.
1fc	3 convolutions + 1 fully connected layer	91.00%+-0.5%	1,532,795
2fc	3 connected layers + 2 fully connected layers	90.78%+-1.2%	11,148,789
4conv	4 convolutions + 3 fully connected layers	91.08%+-1.4%	7,741,965
1fc4conv	4 convolutions + 1 fully connected layers	90.98%+-0.8%	3,292,161
2fc4conv	4 convolutions + 2 fully connected layers	90.68%+-1.12%	7,479,258

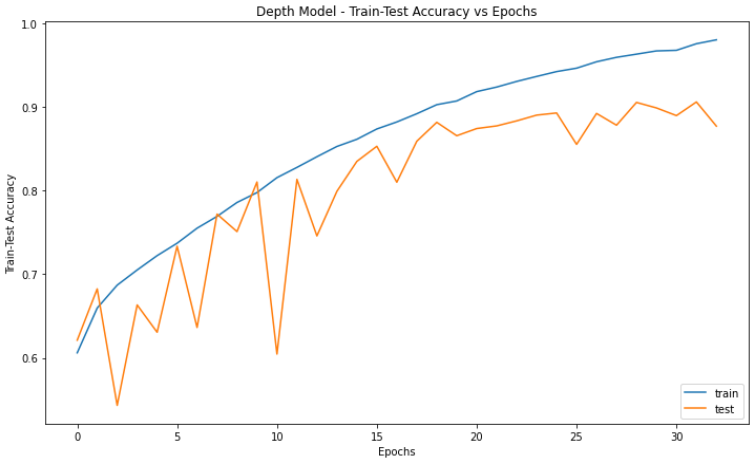

Fig 5.2.1 Train-Test Accuracy vs Epochs

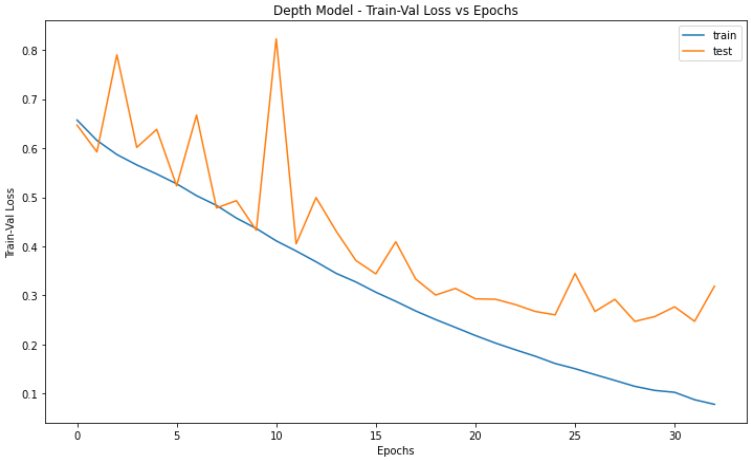

Fig 5.2.2 Train-Validation Loss vs Epochs

With reference to Figs 5.2.1 and 5.2.2, the train accuracy increases steadily towards 1.0 while the train loss decreases steadily towards 0. The test accuracy is not as high due to the limitations of our computation resources, which limited our number of epochs.

5.3 Experiment on Width

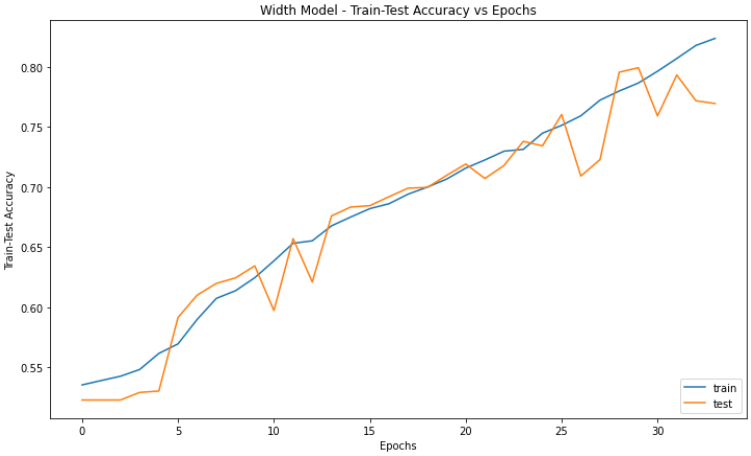

Fig 5.3.1 Train-Test Accuracy vs Epochs

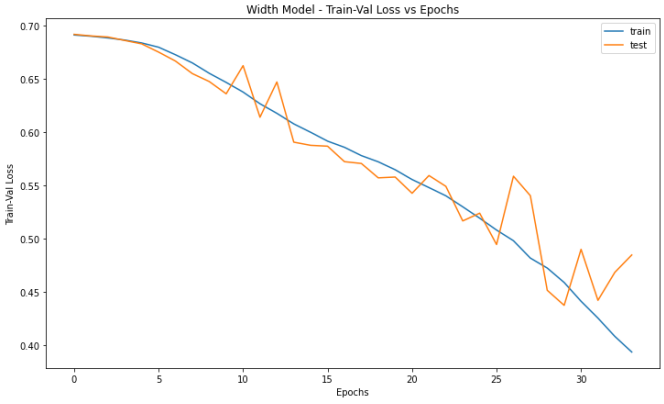

Fig 5.3.2 Train-Validation Loss vs Epochs

With reference to Figs 5.3.1 and 5.3.2, the accuracies and losses are similar to the ones above in Figs 5.2.1 and 5.2.2, with the difference being that the fluctuations are not as large.

5.4 Experiment on Dropout

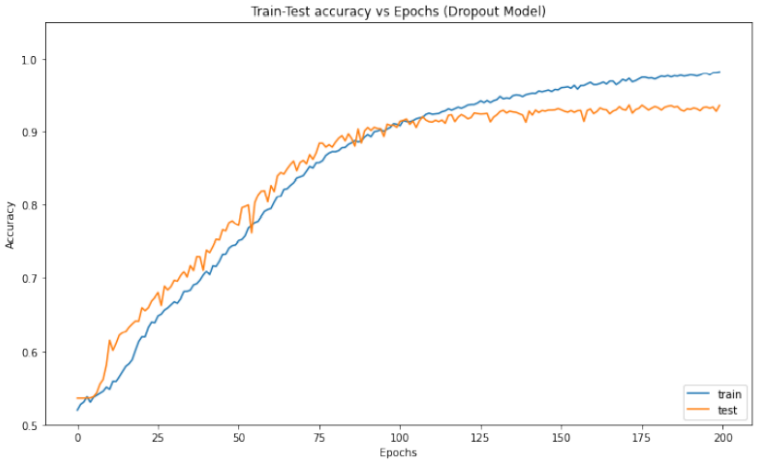

Fig 5.4.1 Train and Test Accuracy vs Epoch

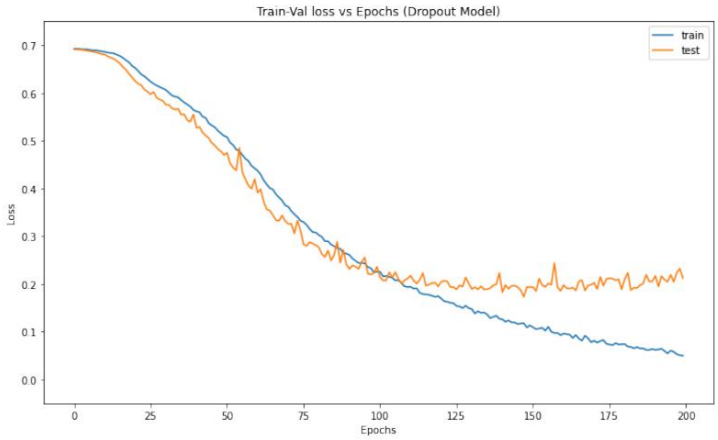

Fig 5.4.2 Train and Validation loss vs Epoch

5.5 Experiment on Optimizer

After multiple attempts, we concluded that RMSProp optimiser yielded the best results.

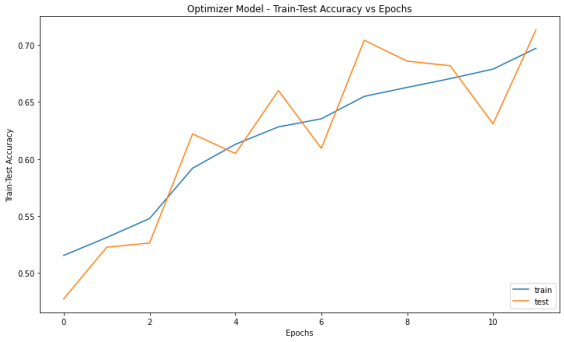

Fig 5.5.1 Train and test Accuracies vs epoch

With reference to Fig 5.5.1, the training accuracy does not show any significant signs of under or over fitting and is a steady climb towards a higher accuracy whereas for the testing accuracy, it is safe to assume that over fitting is present due to an overall increase in accuracy, although with the presence of spikes represented by the occasional drop in accuracies.

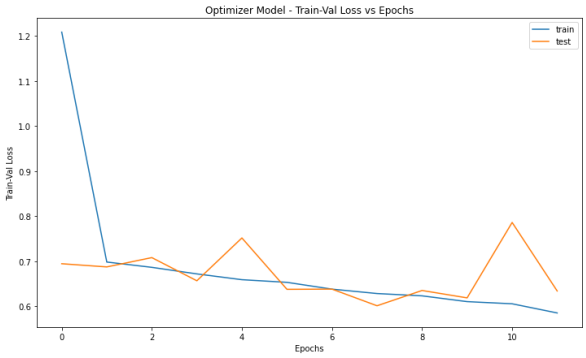

Fig 5.5.2 Train and Validation loss vs Epochs

Like the observations for Fig 5.5.2, the loss rate is at a smooth decline for the training data while it is turbulent for the testing data, which further enhances the theory that over fitting is present for the test data set. Losses are marginally reduced in this result as compared to the optimizer results as its main purpose is to reduce the loss. Hence, we can conclude that RMSProp is the best for this dataset.

6. Conclusion

Overall, from what we have observed, the Adience dataset requires a much higher epoch count to reach saturation point as compared to the CelebA dataset. This is particularly observed when we let the base-line model run for 94 epochs before early stopping was activated and stopped the model. If not, it would have continued until epoch 200.

Due to computation limitations, we are unable to run such a high number of epochs for the other test cases and as such their accuracy rates are much lower as compared to the baseline model. It is also worth noting, that for models that had a higher epoch count they also had a higher accuracy and lower loss rate. Hence, we can deduce that epoch count is a key factor in the accuracy and loss rate for Adience dataset.

CelebA dataset does not experience this as they use transfer learning, and the training data has minimally 800 000 images to work with, which also proves the theory of more data yields better performance. Moving forward, we hope to implement a higher epoch count for data sets such as the Adience dataset without the restriction of our processing units.

7. Reference

Gil Levi, & Tal Hassner. (n.d.). (thesis). *Age and Gender Classification using Convolutional Neural Networks*. Retrieved November 15, 2021, from https://www.cv-foundation.org/openaccess/content_cvpr_workshops_2015/W08/papers/Levi_Age_and_Gender_2015_CVPR_paper.pdf.

SHAKIR, Y. A. S. I. R. H. U. S. S. E. I. N. (2021, September 2). *Gender classification using VGG16+CNN*. Kaggle. Retrieved November 15, 2021, from https://www.kaggle.com/yasserhessein/gender-classification-using-vgg16-cnn/notebook.

Thakur, R. (2020, November 24). *Step by step VGG16 implementation in Keras for beginners*. Medium. Retrieved November 15, 2021, from https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c.

Jermain, N. (2019, April 23). *Gender inference with Deep Learning*. Medium. Retrieved November 15,2021,from https://towardsdatascience.com/gender-identification-with-deep-learning-ac379f85a790.