Examination of Facial Attribute Recognition methods using Transfer Learning on DenseNet with the CelebA Dataset

Sam Zeng[1]

[1] *University of Sydney, Sydney, Australia*

November 16, 2021

1 Introduction

Facial attribute recognition is a commonly studied area in image classification (Liu, 2015), with significant amounts of prior research and data available. This report details the experimentation and fine tuning of a multi-label classification neural network (Maiza, 2019) optimised for facial attribute recognition over the public domain dataset "CelebA", featuring over 200,000 cropped photos of celebrity faces. Each face has been labelled with their respective identified features representing a multi-label dataset with 40 classes.

The model utilises transfer learning on the DenseNet model (Huang, 2016) using the pre-trained "ImageNet" weights, followed by a fine-tuning step unfreezing the top 8 base model layers and retraining it on an even lower learning rate in order to derive a high-accuracy facial attribute recognition system.

The experiment finds a significantly high base accuracy in the base transfer learning case of simply replacing the output layer with a 40-neuron sigmoid activation, indicating that the DenseNet model pre-trained on ImageNet is already highly suited to detecting facial features correctly. Adjusting optimal training parameters of batch size, number of epochs and learning rates, implementing more complex final layers and applying a further fine-tuning step yields marginal increases in accuracy to the classifier.

2 Methods

As the primary purpose of this experimentation was to devise the highest accuracy framework of predicting facial attributes, the process of developing this model was very much through trial and error to observe the optimal model structure, by using elements sourced from prior art, and the hyperparameters were adjusted and fine tuned to produce the highest accuracy based on trials conducted. Thus, whilst the actual experimentation process did not yield a constant linear performance gain per iteration, for the purposes of the findings and model ablation studies the results will be presented from the perspective of ablation, beginning with the highest quality model and systematically breaking down the impacts in accuracy of removing components of the model.

In terms of the model, the highest accuracy produced was through utilising transfer learning on the DenseNet model pre-trained on ImageNet weights. The model structure therefore relies on a DenseNet base, with the top few layers post global average pooling and batch normalisation consisting of an 0.5 dropout layer, 2 fully connected layers of 1024 and 512 neurons, a final 0.5 dropout and then the output activation layer. This model is trained on the dataset before a final fine-tuning step, where the top 8 layers of the DenseNet model are also unfrozen, and the entire model is retrained again on a lower learning rate.

The training epochs were selected by training the model over a larger amount of epochs and then finding where the accuracy of the model peaked, or began to plateau. They were then retrained at only this value. For the optimal model, these were a 10 epoch initial training, followed by a 15 epoch fine-tuning.

The dropout layers are intended to provide further accuracy to the model as it prevents overfitting and unbiases the data. The two fully connected layers serve to gain accuracy as it provides more neurons to make inferences based on the outputs of the DenseNet mode. The final activation layer is made of 40 neurons matching the 40 attribute labels in the dataset. Sigmoid activation was chosen as the labels are all either -1 or 1, but as in the preprocessing stage the -1's are all transformed to zeros, the final output being sigmoid is appropriate as it takes all inputs and outputs as values between 0 and 1, which represents a confidence or probability of each attribute to be 0 or 1. The fine-tuning step also further improves the final model by better adjusting weightings to provide further accuracy.

For the data preprocessing, due to the immensely large volume present in the dataset, there was no augmentation to randomly flip or adjust pictures as is common in smaller sample sets. Rather, all images were resized to square 224x224 resolution and normalised to have pixel values of float ranges between 0 and 1. This is to satisfy the inputs required for the DenseNet model. Additionally, labels were transformed from 1, -1, to 1, 0 to be more in line with traditional binary classification. Finally, the images and labels were combined and then processed into the tensorflow data structure, tf.data, to then be split by the official marked train, evaluate and test sets. These were then batched in sizes of either 64 or 32, dependent on the experimentation. All training took place locally on a Nvidia RTX2070 GPU.

3 Experimental Setup

As mentioned above, the dataset was sufficiently large as to not require randomised augmentation. They were simply normalised and squared to fit the inputs of the DenseNet model.

With regards to evaluation, binary accuracy was chosen as it is calculated by how often predictions matched binary labels, supporting the multi-label classification problem that this project entails. This metric is by far the most suitable for evaluation of this project as it provides the fairest assessment and weighting of both correct and incorrect labels to which there are a great number of (TensorFlow, 2021). Thus, when considering this problem as a binary multi-label classification problem, the averaged accuracy which is demanded by the specifications is best represented through this metric.

4 Results & Discussion

The final model produced evaluation accuracies of 89.09%, with the optimal hyperparameters found to be a batch size of 64, running for 10 epochs initially at a learning rate of 0.001 and then fine-tuning the model on another model for a further 15 epochs at a learning rate of 0.00001 with the top 8 layers of DenseNet unfrozen. While this accuracy is high, this is only a marginal gain over optimising hyperparameters and refitting a DenseNet model to output a 40 neuron sigmoid activation, which produced an accuracy of 84.79%. A table below displays the accuracies of each method employed, along with notes detailing the components ablated contributing to the accuracy loss.

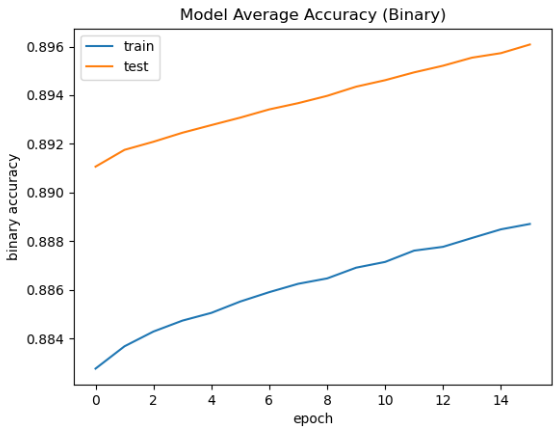
Note that while the actual experimental trials were not conducted in this order, for the purposes of the report and ablation studies, the models will be presented in descending order of accuracy and contextualised as an ablation study to show the contributions to accuracy of particular components within the model.

Ranking	Batch Size	Epochs	Learning Rates	Accuracy (Binary)
1	64	25 (10+15)	0.001/0.00001	89.09%
2	64	20 (10+10)	0.001/0.00001	88.96%
3	64	20	0.0001	88.70%
4	32	20	0.0001	88.64%
5	32	13	0.0001	88.42%
6	32	20	0.01	86.85%
7	32	13	0.0001	86.79%
8	32	20	0.0001	86.49%
9	32	13	0.0001	84.79%
10	32	30	0.0001	82.99%

Over the 10 trials, the total difference in accuracy of the best and worst models was 6.1%. All components attempted provided slight marginal benefits. Below is a summary of the differences in method for all 10 trials.

As outlined in the method, the most accurate model, yielding an accuracy of 89.09%, consisted of all components, including the 15 epoch, 0.00001 learning rate fine-tuning step, 2 dropout layers and 2 fully connected layers of 1024 and 512 neurons before the output, with a batch size of 64 and 10 initial epochs at 0.001 learning rate.

The second highest model was equivalent to this one, without as many epochs of fine-tuning. This resulted in a slightly lower total accuracy of about 0.13%, which can be attributed to the model learning not peaking yet, as seen in the diagram below of the learning rates of the optimal model, which peak just at around 15 epochs. The 10 epoch fine-tuning does not yet reach the peak.

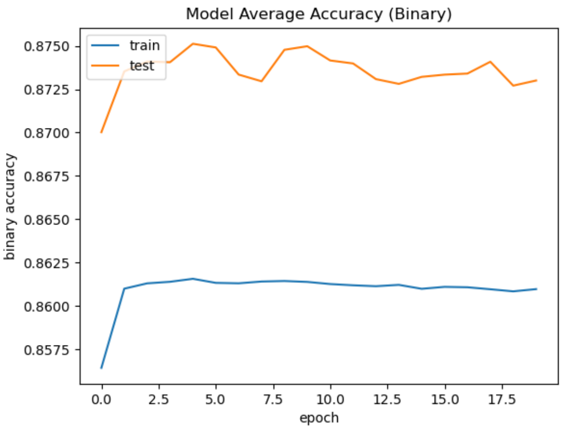


The third model consists of the optimal model without the final fine-tuning step. We see that this model predicts at an accuracy of 88.70%, which is 0.39% lower than optimal. From this we can gather that the fine-tuning step equates to about a 0.39% extra accuracy to the final model.

The fourth model tests a lower batch size of 32 instead of 64, which equates to a 0.06% drop in accuracy. This is marginal, and reflects a mostly similar model accuracy when modifying parameters such as this.

The fifth model is a similar test to the second, running the above model with a smaller training epoch size. This equates to an accuracy drop of 0.22%, and again reflects the model not yet reaching its peak accuracy yet.

The sixth model tests the effects of learning rate adjustments, with a higher learning rate resulting in a greater fluctuation in predictions as it tries to jump to conclusions at a faster rate, shown in the following diagram. This results in a lower final accuracy of 86.79%, which is 2.3% lower than the optimal and thus choosing a lower learning rate was significant in improving model performance.



Finally, the last few models are further ablative studies on components of the final layers, with optimal learning rates set. Thus, for the following comparisons, the non-fine-tuned model will be used as the baseline to evaluate the decreases in performance as a result of ablating particular components. Additionally, due to the memory intensity of training with batch size 64 and a neglible difference between 64 and 32 batch size, the following were trained on batch sizes of 32, yielding a baseline accuracy of comparison of effectively 88.64%.

Without dropouts, this accuracy dropped to 86.79%, a reduction of 1.85% accuracy. Not having the 512 neuron fully connected layer resulted in a further 0.3% drop in accuracy, and finally without the 1024 FC layer, there was a substantial further 3.5% drop in accuracy. Thus, the removing the constructed top layers resulted in a combined 5.65% drop in accuracy. It should be noted however, that in the worst performing model, the optimal epoch was found to be different as the rate of learning spiked faster and then decreased over the standard 20 epoch test, and the peak accuracy was actually 84.79%, which then re-evaluates the drop as 1.7% and a combined 3.85% drop in accuracy, 4.3% lower than the optimal model. These results are summarised in the table below.

Ablation	Change From Previous	Cumulative Decrease
None	0%	0%
No Fine-tuning	0.39%	0.39%
Reduced Batch Size	0.06%	0.45%
No Dropouts	1.85%	2.3%
No 512-FC Layer	0.3%	2.6%
No 1024-FC Layer (optimised)	1.7%	4.3%

5 Conclusion

From these results we can see that the biggest contributors to improvements in accuracy as compared with the base DenseNet model were applying a 1024 neuron FC layer before the output and adding dropouts. The smallest gain (with exception to the batch size change) was applying an additional 512 neuron FC layer before the output layer. The trial adjustment to hyperparameters such as epoch limit and learning rates affected the output accuracy in similar magnitudes to these components.

Overall however, even the basic DenseNet model provides a significantly high accuracy of 84.79%, with the optimal model improving on that by a marginal 4.3% to yield 89.09% accuracy.

References

Huang G., Liu Z. Maaten L. (2016). “Densely Connected Convolutional Networks”. In: *Conference on Computer Vision and Pattern Recognition*. URL: `https://arxiv.org/abs/1608.06993`.

Liu Z., Luo P. Wang X. Tang X. (2015). “Deep Learning Face Attributes in the Wild”. In: *International Conference on Computer Vision*. URL: `https://arxiv.org/abs/1411.7766`.

Maiza, A (2019). "Multi-Label Image Classification in TensorFlow 2.0". In: *Review of Scientific Instruments*. URL: `https://towardsdatascience.com/multi-label-image-classification-in-tensorflow-2-0-7d4cf8a4bc72`.

TensorFlow (2021). "TensorFlow". In: *TensorFlow API Docs*. URL: `https://www.tensorflow.org/api_docs/python/tf`.