Optimizing Facial Attribute Classification Models: From Baseline to Enhanced Architectures

Chenuka Garusinghe

November 3, 2024

This article focuses on Facial attribute extraction, a Project done using ResNet-18 initially as a baseline and moving on to more advanced techniques and models like EfficientNet-B0 with Weighted Focal Loss for better training and model performance.

1 Introduction

We used the widely available CelebA dataset to develop the deep learning models for multi-label facial attributes. Starting off with baseline model employing pretrained ResNet-18 architecture, we progressively introduce enhancements to address identified limitations. Finally we proceeded with several data augmentations, weighted focal loss function to improve the classification performance.

The findings show the effects of the modifcations, loss function adjustments, and data augmentation significantly enhance model accuracy and generalization capabilities. This comparison helped decied the final model.

2 Methods

2.1 Baseline Model (Model 1)

Architecture: In this first model we decided to go with the **RestNet-18** architecture from the pytorch library (specifically torchvision models) and in order to improve the model it was decided to leverage and transfer learning from the ImageNet dataset. This is so that knowledge about general features in images, like edges, texture and shape are better understood. This allows us to Fine tune the model on CelebA.

Data Processing:

- **Transformations:** We decided to use torchvision.transforms for basic image pre-processing. We used this to convert images to PyTorch tensors, and followed to standardize pixel values with a mean 0.5 and a standard deviation of each channel.

Training Setup:

- **Loss Function:** `Binary Cross-Entropy with Logits Loss` is used be because it is particualrly suited for multilabeled binary classification tasks whic can allow the model to independently predict the rpesence or absence of each attributes, ideal for the celebA dataset.

- **Optimizer:** It was decieded to incooperate **Adam optimizer** with a learning rate of 0.001, because of its typically leading towards a faster convergence and better generalisation. Having this moderatley low learning rate will allow us to convereg at a steady space reducing the risk of overshooting lcoal minimum during the training phase.

- **Mixed Precision Training:** Due to computational limitations the epoch count was reduced to 10 but inroder to prevent any further degradation to the training, we utilized mixed precision training via PyTorch's autocast. This whilist Leveraging Metal Performace Shaders (MPS) allowed us to iuse lower precisino (float16) for certain parts and higher precision (float32) where needed dynamically maintaing accuracy without getting bottle necked by speed.

2.2 Enhanced Model with Data Augmentation (Model 2)

Architecture:
While the final model primarily builds on the baseline model, a key change is the use of the pre-trained EfficientNet-B0 model. The replacement of the default clasiffer is also something to note. The cusotm classifer head was defined such that a Shared feature block that helps reduce th edimmesional output of the backbone of this mode is scaled down for more computationally manageable units of 512, followed with a batch normalization on the output of the linear layer. This will help reduce any internal co-variate shifts, that'll help the network to better for new distribution of data in coming batches. Additionally a Rectified Linear Unit (ReLU) function was introduced for non linearity purposes so the model can lean more complex patterns and interactions among features, which are common among facial attributes. To help prevent over fitting however, a dropout is added to randomly drop a fraction of neurons, so the network does not rely too heavily on any single feature other than pattern.

Data Processing:

- **Transformations:** It was decieded to focus a bit more heavily on the tranisformations in hand in this model, due to the nature of the dataset and for better coverage. The Training augmentations, it follows a horizontal flip, an addition of Gaussian Noise (to simulate real world image noise), the use of ImageNet meand and S.D strictly for consitency purposes and as earlier normalizations to ensure evaluation consitency.

  It is also worth metioning that the use of the Attribute Processor is to tackle the possible imbalances in the atrribiutes of the CelebA dataset. Because in a datasets like CelebA it is possible that certain facial attributes may be highly imbalanced; for example, some features like "Smiling" may appear frequently, while others like "Bald" may be rare. Keeping track of these as Positive ratios (frequency of each attribute), Weights for Loss Adjustments and more can help streamline training process.

Training Setup: Broadly we followed the setup as is in the training of the baseline model, but the few key changes that showed improvement is discussed as below:

- **Learning Rate Scheduler: OneCycleLR** We used this with a learning rate of 3×10^{-4}, slowly reducing it as we train the model from the initial increase. The dynamic adjustment allows the model to explore more wider ranges of learing rates, possibly preventing any sharp local minima and achieve better unbiased convergence. This is fitted along with a early stopping mechanism so that if the model's validation loss fails to improve over a set number of consecutive epochs (patience threshold), training halts automatically, so that the model isnt trained in vain.

3 Experimental Setup

3.1 Dataset

We used the CelebA dataset with over 200,000 celebrity images with anotation on 40 binary facial attributes. While certiain attributes like attractiveness can be debated on the fareness of using a binary attribute, it is something that was looked over for this model in particualr. Regardless the iamges represented a wide range of facial expressions, poses etc...

The data was split to approximately 162,770 images with the validation set containing approximately 19,867 images. The test set saw a higher set of images comapred to the validation set by an additional 100 images, ustilized in the evaluation.

It was decided to convert attribute labels from -1/1 to 0/1 to represent binary classes properly and using the custom dataset classes we handled image loading more efficiently.

4 Results and Discussion

4.1 Model Performance Comparison

After testing both models there were slight notable improvements in certain metrics in the final model compared to the baseline. In this section we present and discuss the evaluation results for each model, while looking into on improvements observed through architectural changes, data augmentation, and custom training strategies

4.1.1 Comparative Performance Analysis

The performance of **model_1** (baseline) and the **final_model** (optimized) on selected facial attributes is as below. Here we highlight the accuracy, precision, recall, and F1 scores to get a well rounded overview of the each models characteristics.

- **Accuracy**: The **final_model** achieved a marginally higher accuracy (88.92%) when it was compared to **model_1** (88.14%), showing an improvement in correctly predicted instances overall.

- **Precision**: Notably, the **final_model** shows a significant increase in the precision (85.60%), meaning that there was a higher rate of correct prediction across all the postives that the model was able to identify. This improvements is significantly reflected on attributes like *Attractiveness* and *Eye-Glasses*.

- **Recall**: A significant drop was seen here, highlighting that the final model missed actual postive cases, despite the better precision, meaning that the final model became for selective and precise

- **F1 Score**: Because the F1 Score balances out the precision and recall results we saw a decrease in this metric for the final model too overall.

4.1.2 Attribute-Specific Performance

The goal of this section is to highlight some specific attributes that in general, most models that were tested prior to this, that sits between the baseline model and the final model find challenging to predict and have large variance between each others results when done so.

- **Attractive**: We see that because of the higher selectiveness of the final model allowed us to better determine this attribute. The final model achieved a higher F1 score (0.7467) compared to model 1 (0.681)

- **Eyeglasses**: Both models were quite similar in results in this attribute with final model slightly out-performing model 1 (0.9389 vs. 0.9271) in F1 score, leading to believe that this is likely due to advantage of the increased depth in the Neural Network.

- **5_o_Clock_Shadow**: We saw a decrease in the performance in this attribute in the final model by nearly 18% suggesting that the model may have overfitted some complex augmentaions causing it to be less inclusive of subtle features.

- **Young**: Both models performed almost identically in identifying the *Young* attribute, with the**final_model** slightly improving the F1 score (0.8977) over **model_1** (0.8953).

4.1.3 Visualization and Training Progress

We can see the visualization of the training process, the distribution of the batch loss and and learning rate schedule for each model down below.

- **Training and Validation Loss** (Fig. 2): We can see that despite the shortcomings mentioned in the final model the training and validation loss curves show that the **final_model** scores a lower loss value, meaning better convergence. The earlier mentioned early stopping mechanism played a crucial role in the training as seen the figure, stop at the 6th Epoch preventing over fitting , while the first model continued to train till all 10.

Figure 1: *Training and Validation Loss Curves Model 1*

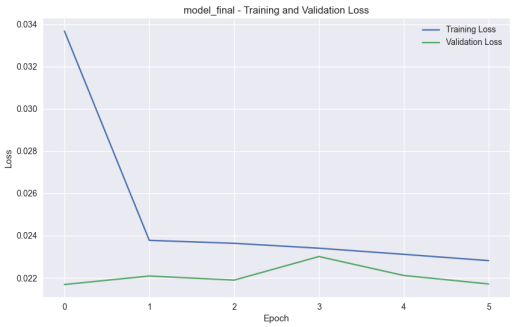

Figure 2: *Training and Validation Loss Curves Final Model*

- **Batch Loss Distribution** (Fig. 4): When looking at the first models batch loss, it seems to be slightly skewed. Regardless in compression to the final model, we see that the batch loss distribution of the first model is much more wider, indicating variability in the model's performance across batches. This means that the final model is more consistent and stable in the training process, an attribute of the above mentioned augmentations. This allows the final model to minimise outlier losses and optimise for generalisation

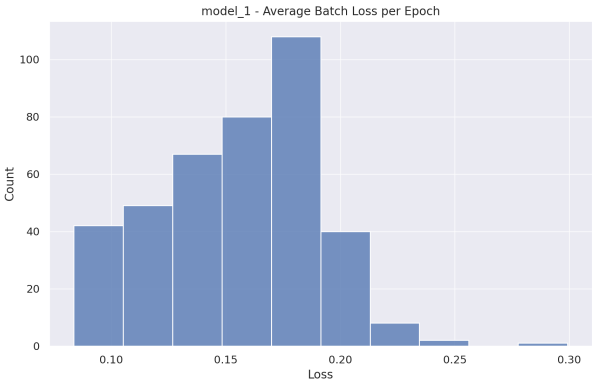

Figure 3: *Batch Loss Distribution for Model 1*

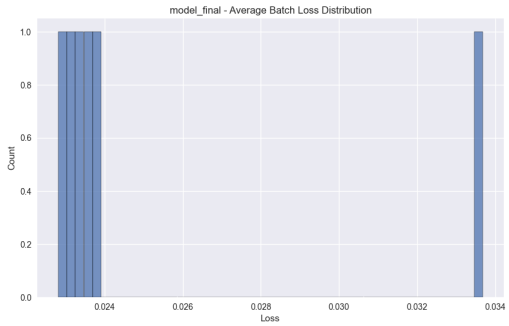

Figure 4: *Batch Loss Distribution for Final Model*

- **Learning Rate Schedule** (Fig. 5): We can see that the learning rate dynamically is adjusted in the final model allowing for optimized convergence, where as the baseline model used a static learning rate, leading to a slower convergence overall.

Figure 5: *Learning Rate Schedule for Model 1*

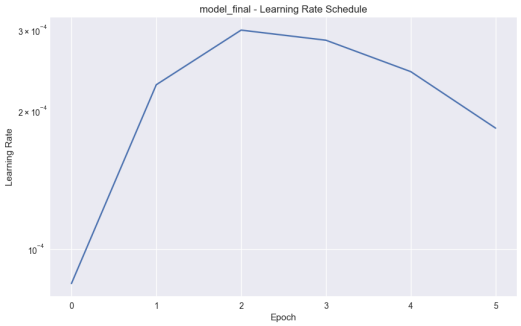

Figure 6: *Epoch Training Times for Final Model*

4.1.4 Discussion of Model Improvements and Limitations

While th final model did incopperate some key advanced techniques such as using EfficintNet as the backbone and several data augmentations with weighted focal loss the trade off in the gain for Accurracy and Precission were reflected in the above metrics, specifically recall. This leaves imporvement to lean towards some features of the baseline model, so that the last and final model will no longer beecome more selective and less inclusive. We can also go onto fine tune the hyper parameters of focal loss in future models to address the rare attributes.

5 Conclusion

Overall the study showed some improvements that data augmentation, using Weighted Focal Loss and managing learning rate more efficiently does lead to significant outcomes of the models and these are some things to consider in future progressions regardless of the base model used.

6 References

References

[1] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

[2] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and Flexible Image Augmentations," *Information*, vol. 11, no. 2, p. 125, 2020.

[3] L. Liu, Z. Luo, X. Wang, and X. Tang, "CelebA: Large-scale CelebFaces Attributes Dataset." Available: http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html