Introduction

In this homework, we address the task of gender classification using transfer learning. Our aim is to create a model that can accurately determine the gender of individuals based on their facial images and specific labels. To achieve this, we utilize the approach of transfer learning, which involves making use of the existing knowledge and capabilities of pre-trained models. By adapting the pre-trained VGG-16 model to our specific task, we aim to construct a gender classification model that can effectively generalize unseen data.

Dataset

The dataset used for this task is a subset of the CelebA dataset. It consists of a diverse collection of 30,000 facial images, with each image having 10 different labels (Male Blond_Hair, Eyeglasses, Wearing_Earrings, Bangs, Young, Smiling, Heavy_Makeup, Straight_Hair, Black_Hair). Here are the randomly selected five images and displayed them along with their respective gender labels:

Image 3:

→ “Male” column is 1

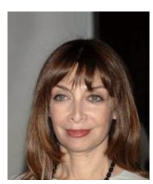
Image 30:

→ “Male” column is 1

Image 300:

→ “Male” column is 1

Image 3000:



→ “Male” column is 0

Image 30000:

→ “Male” column is 1

Methodology

To accomplish the gender classification task, we adopted the transfer learning approach using the VGG-16 model. The VGG-16 model is a deep convolutional neural network pre-trained on the ImageNet dataset, consisting of images from various categories.

To use the pre-trained VGG-16 model, we made the following adaptations:

- Feature Extraction: We used the pre-trained VGG-16 model as a feature extractor by removing the last classification layer (top layer) and retaining the bottleneck layer. The bottleneck layer captures more general features and exhibits better transferability compared to the final layer.

- Freezing the Base Model: We set all layers in the base model to be non-trainable. By freezing the base model, we ensure that the learned weights from the ImageNet dataset remain constant and prevent them from being updated during training.

- Binary Classification Head: We added a binary classification head on top of the base model. This additional layer consists of a flatten operation followed by fully connected layers with ReLU activation. The final layer uses the sigmoid activation function to output a probability value for gender classification.

By combining the base model and the binary classification head, we created a new gender classification model capable of predicting the gender of individuals from facial images.

Experiments

In the first part of the experiments, I did an analysis to determine the optimal learning rate for the gender classification model. I performed a trial run with and without fine-tuning using three different learning rates: 0.1, 0.001, and 0.00001, training the model for three epochs with each learning rate.

I used various techniques to optimize the performance and prevent overfitting of our gender classification model. Initially, I implemented early stopping and learning rate reduction strategies during the training process. Early stopping helps prevent overfitting by keeping track of the validation loss and terminating training when the loss fails to improve. Learning rate reduction adjusts the learning rate during training to fine-tune the model's convergence.

After evaluating the results of the initial experiments, I used additional techniques to further enhance the model's performance. Specifically, I implemented dropout and batch normalization layers. Dropout randomly sets a fraction of input units to zero during training, preventing over-reliance on specific features and improving generalization. Batch normalization normalizes the inputs across the mini-batch, reducing internal covariate shift and accelerating training.

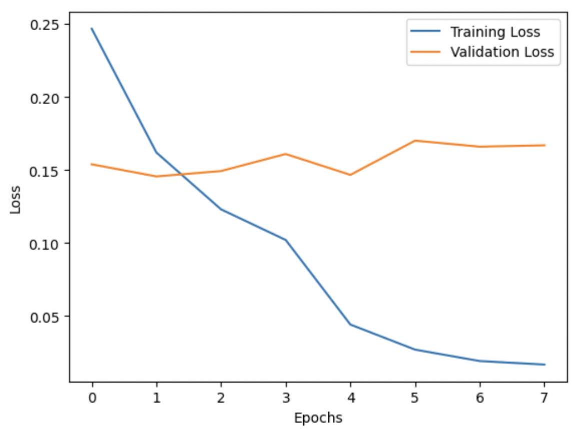
In the below results section, there are the findings of the initial experiments. As it can be seen, without fine-tuning, the best test accuracy was achieved using a learning rate of 0.001. However, upon fine-tuning the model, it is discovered that a learning rate of 0.00001 yielded the highest test accuracy. These results indicate the effectiveness of different learning rates in different scenarios.

Case no:	Fine-tuning	Learning Rate	Train Accuracy	Validation Accuracy	Test Accuracy
1	No	0.1	Epoch 1: 0.8748 Epoch 2: 0.9154 Epoch 3: 0.9317	Epoch 1: 0.9287 Epoch 2: 0.8753 Epoch 3: 0.8917	0.8963
2	No	0.001	Epoch 1: 0.9002 Epoch 2: 0.9383 Epoch 3: 0.9550	Epoch 1: 0.9430 Epoch 2: 0.9437 Epoch 3: 0.9478	0.9416
3	No	0.00001	Epoch 1: 0.8257 Epoch 2: 0.9040 Epoch 3: 0.9277	Epoch 1: 0.9287 Epoch 2: 0.9393 Epoch 3: 0.9445	0.9383
4	Yes	0.1	Epoch 1: 0.8806 Epoch 2: 0.9180 Epoch 3: 0.9415	Epoch 1: 0.9498 Epoch 2: 0.9408 Epoch 3: 0.9350	0.9388
5	Yes	0.001	Epoch 1: 0.9350 Epoch 2: 0.9304 Epoch 3: 0.9326	Epoch 1: 0.9350 Epoch 2: 0.4332 Epoch 3: 0.9413	0.9436
6	Yes	0.00001	Epoch 1: 0.8305 Epoch 2: 0.9102 Epoch 3: 0.9319	Epoch 1: 0.9267 Epoch 2: 0.9423 Epoch 3: 0.9478	0.9484

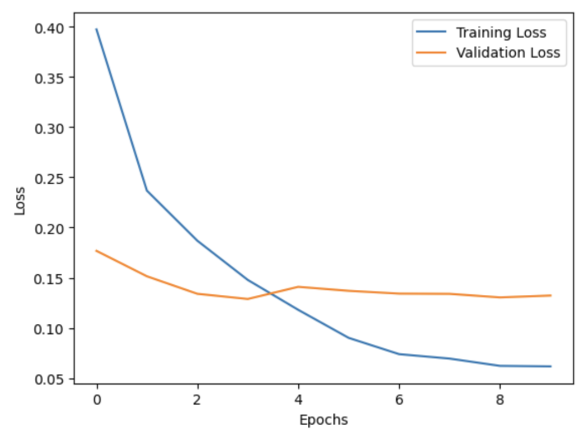
Subsequently, based on the chosen learning rates, I proceeded to train the two models (without fine-tuning with a LR of 0.001, and with fine-tuning with a LR of 0.00001) for an extended period of ten epochs to further refine their performances. The results, including the accuracies achieved and the training curve depicting the training and validation loss over epochs, are summarized in the table and figures below.

Case no:	Fine-tuning	Learning Rate	Train Accuracy	Validation Accuracy	Test Accuracy
1	No	0.001	Epoch 1: 0.9027 Epoch 2: 0.9397 Epoch 3: 0.9551 Epoch 4: 0.9635 Epoch 5: 0.9852 Epoch 6: 0.9919 Epoch 7: 0.9950 Epoch 8: 0.9953 Early stopping	Epoch 1: 0.9388 Epoch 2: 0.9432 Epoch 3: 0.9423 Epoch 4: 0.9453 Epoch 5: 0.9483 Epoch 6: 0.9468 Epoch 7: 0.9470 Epoch 8: 0.9458 Early stopping	0.9428
2	Yes	0.00001	Epoch 1: 0.8277 Epoch 2: 0.9061 Epoch 3: 0.9286 Epoch 4: 0.9445 Epoch 5: 0.9559 Epoch 6: 0.9662 Epoch 7: 0.9732 Epoch 8: 0.9758 Epoch 9: 0.9780 Epoch 10: 0.9780 Early stopping	Epoch 1: 0.9310 Epoch 2: 0.9432 Epoch 3: 0.9477 Epoch 4: 0.9570 Epoch 5: 0.9473 Epoch 6: 0.9507 Epoch 7: 0.9537 Epoch 8: 0.9547 Epoch 9: 0.9553 Epoch 10: 0.9560 Early stopping	0.9526

Case no: 1, train-val loss versus epoch:



Case no: 2, train-val loss versus epoch:



Conclusion

In conclusion, I evaluated the performance of our gender classification model on the test set using various configurations. I found that fine-tuning the model with a learning rate of 0.00001 resulted in the highest accuracy. By using techniques such as early stopping, learning rate reduction, dropout, and batch normalization, I tried to prevent overfitting and improved generalization capabilities. The model achieved notable accuracy on the test set, showing the ability to accurately classify gender based on images. The results highlighted the significance of fine-tuning and the impact of different techniques in enhancing the performance of gender classification models.