

https://colab.research.google.com/drive/1C_V4aZ8V5qxMerplG1_wMll0FWt5cqG8?usp=sharing

Introduction:

The problem is to be able to correctly identify whether a given image is female or male. In order to complete this task, a large-sized dataset will be used to train the model.

Dataset:

The dataset is CelebA dataset. It contains more than 200k celebrity face images with 40 attributes. 30k of these images are used with gender attribute. Images are resized to 224 x 224, also has RGB values (i.e input shape is 224x224x3). 30k images are divided into train (24k), validation (3k) and test (3k) data (each with 2 labels).

Methodology:

Our task is to build a image classifier using Keras library. A pretrained VGG16 model is used as the backbone of the model, then manually crafted dense layer is added in order to correctly classify into 2 categories: male and female. Layers of the base model is frozen in order to make sure weights are constant. The dense layer has 2 output nodes, therefore softmax activation function and categorical cross entropy loss function is used.

Experiments:

The hyperparameters test during the experiments are batch size, number of epochs and learning rate. Validation accuracy is used as the prime factor while deciding the hyperparameters.

Experiment 1: batch_size = 8, n_epochs = 10, learning_rate = 0.001

```
Epoch 1/10
1500/1500 [=====] - 171s 114ms/step - loss: 11.1020 - accuracy: 0.9092 - val_loss: 7.8448 - val_accuracy: 0.9240
Epoch 2/10
1500/1500 [=====] - 123s 82ms/step - loss: 5.7641 - accuracy: 0.9426 - val_loss: 9.6034 - val_accuracy: 0.9307
Epoch 3/10
1500/1500 [=====] - 115s 77ms/step - loss: 3.5407 - accuracy: 0.9618 - val_loss: 7.8265 - val_accuracy: 0.9397
Epoch 4/10
1500/1500 [=====] - 115s 77ms/step - loss: 2.8420 - accuracy: 0.9653 - val_loss: 9.5461 - val_accuracy: 0.9240
Epoch 5/10
1500/1500 [=====] - 116s 77ms/step - loss: 1.8017 - accuracy: 0.9738 - val_loss: 9.3972 - val_accuracy: 0.9250
Epoch 6/10
1500/1500 [=====] - 123s 82ms/step - loss: 1.5229 - accuracy: 0.9754 - val_loss: 13.4987 - val_accuracy: 0.9053
Epoch 7/10
1500/1500 [=====] - 123s 82ms/step - loss: 0.8837 - accuracy: 0.9834 - val_loss: 7.7629 - val_accuracy: 0.9360
Epoch 8/10
1500/1500 [=====] - 123s 82ms/step - loss: 0.6518 - accuracy: 0.9869 - val_loss: 8.0141 - val_accuracy: 0.9373
Epoch 9/10
1500/1500 [=====] - 123s 82ms/step - loss: 0.5079 - accuracy: 0.9885 - val_loss: 8.2472 - val_accuracy: 0.9387
Epoch 10/10
1500/1500 [=====] - 123s 82ms/step - loss: 0.3909 - accuracy: 0.9904 - val_loss: 8.5161 - val_accuracy: 0.9370
```

Even though accuracy peaked at 3rd epoch, it can be said that after some point (7th epoch and onwards) accuracy converges around 93.75

Experiment 2: batch_size = 64, n_epochs = 10, learning_rate = 0.001

```
Epoch 1/10
1500/1500 [=====] - 116s 77ms/step - loss: 11.0126 - accuracy: 0.9099 - val_loss: 7.4603 - val_accuracy: 0.9317
Epoch 2/10
1500/1500 [=====] - 123s 82ms/step - loss: 5.5405 - accuracy: 0.9456 - val_loss: 8.4016 - val_accuracy: 0.9293
Epoch 3/10
1500/1500 [=====] - 123s 82ms/step - loss: 3.6778 - accuracy: 0.9586 - val_loss: 7.4906 - val_accuracy: 0.9407
Epoch 4/10
1500/1500 [=====] - 123s 82ms/step - loss: 2.7868 - accuracy: 0.9650 - val_loss: 13.8301 - val_accuracy: 0.9100
Epoch 5/10
1500/1500 [=====] - 123s 82ms/step - loss: 1.8698 - accuracy: 0.9736 - val_loss: 9.1623 - val_accuracy: 0.9367
Epoch 6/10
1500/1500 [=====] - 115s 76ms/step - loss: 1.2829 - accuracy: 0.9800 - val_loss: 8.3585 - val_accuracy: 0.9370
Epoch 7/10
1500/1500 [=====] - 116s 77ms/step - loss: 0.9605 - accuracy: 0.9825 - val_loss: 8.6747 - val_accuracy: 0.9390
Epoch 8/10
1500/1500 [=====] - 117s 78ms/step - loss: 0.6989 - accuracy: 0.9859 - val_loss: 8.5151 - val_accuracy: 0.9333
Epoch 9/10
1500/1500 [=====] - 117s 78ms/step - loss: 0.5097 - accuracy: 0.9878 - val_loss: 8.8621 - val_accuracy: 0.9363
Epoch 10/10
1500/1500 [=====] - 124s 82ms/step - loss: 0.3978 - accuracy: 0.9900 - val_loss: 8.5498 - val_accuracy: 0.9387
```

Again at 3rd epoch, accuracy peaked to 94. Similarly, after some point (5th epoch onwards this time) accuracy converges around 93.75. Since the initial epochs have higher accuracy, higher peak accuracy and earlier convergence; batch size 64 is preferred over 8 for the model.

Experiment 3: batch_size = 64, n_epochs = 10, learning_rate = 0.0001

```
Epoch 1/10
1500/1500 [=====] - 124s 82ms/step - loss: 1.5417 - accuracy: 0.9043 - val_loss: 1.1976 - val_accuracy: 0.9270
Epoch 2/10
1500/1500 [=====] - 116s 77ms/step - loss: 0.6886 - accuracy: 0.9457 - val_loss: 1.0157 - val_accuracy: 0.9230
Epoch 3/10
1500/1500 [=====] - 116s 77ms/step - loss: 0.4548 - accuracy: 0.9563 - val_loss: 1.0871 - val_accuracy: 0.9280
Epoch 4/10
1500/1500 [=====] - 124s 82ms/step - loss: 0.2619 - accuracy: 0.9684 - val_loss: 1.5365 - val_accuracy: 0.9037
Epoch 5/10
1500/1500 [=====] - 116s 77ms/step - loss: 0.1954 - accuracy: 0.9740 - val_loss: 0.9946 - val_accuracy: 0.9347
Epoch 6/10
1500/1500 [=====] - 116s 77ms/step - loss: 0.1262 - accuracy: 0.9799 - val_loss: 0.9998 - val_accuracy: 0.9330
Epoch 7/10
1500/1500 [=====] - 124s 82ms/step - loss: 0.0875 - accuracy: 0.9851 - val_loss: 0.9824 - val_accuracy: 0.9377
Epoch 8/10
1500/1500 [=====] - 115s 77ms/step - loss: 0.0551 - accuracy: 0.9881 - val_loss: 0.9963 - val_accuracy: 0.9340
Epoch 9/10
1500/1500 [=====] - 116s 77ms/step - loss: 0.0338 - accuracy: 0.9925 - val_loss: 1.0433 - val_accuracy: 0.9357
Epoch 10/10
1500/1500 [=====] - 116s 77ms/step - loss: 0.0184 - accuracy: 0.9954 - val_loss: 1.0295 - val_accuracy: 0.9323
```

Overall accuracy is similar to learning rate of 0.001, but validation loss is lower. Therefore, even though it takes more time to train, 0.0001 is preferred over 0.001.

Experiment 4: batch_size = 64, n_epochs = 5, learning_rate = 0.01

```
Epoch 1/10
1500/1500 [=====] - 117s 78ms/step - loss: 111.7484 - accuracy: 0.9090 - val_loss: 113.1532 - val_accuracy: 0.9177
Epoch 2/10
1500/1500 [=====] - 124s 82ms/step - loss: 58.0100 - accuracy: 0.9434 - val_loss: 71.7617 - val_accuracy: 0.9407
Epoch 3/10
1500/1500 [=====] - 116s 77ms/step - loss: 39.1771 - accuracy: 0.9574 - val_loss: 81.2820 - val_accuracy: 0.9433
Epoch 4/10
1500/1500 [=====] - 115s 76ms/step - loss: 26.4014 - accuracy: 0.9663 - val_loss: 89.2831 - val_accuracy: 0.9353
Epoch 5/10
1500/1500 [=====] - 115s 77ms/step - loss: 16.4051 - accuracy: 0.9756 - val_loss: 79.1272 - val_accuracy: 0.9410
```

As expected, even though accuracy is around same and model trains faster, high learning rate caused very bad validation loss results – meaning the model is not as confident with its results.

Experiment 5: batch_size = 32, n_epochs = 5, learning rate = 0.01

```
Epoch 1/10
1500/1500 [=====] - 117s 78ms/step - loss: 112.0194 - accuracy: 0.9086 - val_loss: 89.7724 - val_accuracy: 0.9290
Epoch 2/10
1500/1500 [=====] - 117s 78ms/step - loss: 56.5216 - accuracy: 0.9468 - val_loss: 75.3775 - val_accuracy: 0.9287
Epoch 3/10
1500/1500 [=====] - 126s 84ms/step - loss: 38.9909 - accuracy: 0.9568 - val_loss: 94.2968 - val_accuracy: 0.9217
Epoch 4/10
1500/1500 [=====] - 124s 82ms/step - loss: 28.1861 - accuracy: 0.9661 - val_loss: 74.4777 - val_accuracy: 0.9420
Epoch 5/10
1500/1500 [=====] - 123s 82ms/step - loss: 21.1673 - accuracy: 0.9721 - val_loss: 80.1570 - val_accuracy: 0.9407
```

Not much changed, but batch_size 64 is overall more consistent, therefore will be preferred.

Experiment 6: batch_size = 64, n_epochs = 20, learning rate = 0.0001

```
Epoch 1/20
1500/1500 [=====] - 130s 80ms/step - loss: 1.5557 - accuracy: 0.9041 - val_loss: 1.1136 - val_accuracy: 0.9173
Epoch 2/20
1500/1500 [=====] - 129s 86ms/step - loss: 0.6791 - accuracy: 0.9458 - val_loss: 1.1823 - val_accuracy: 0.9177
Epoch 3/20
1500/1500 [=====] - 129s 86ms/step - loss: 0.4543 - accuracy: 0.9546 - val_loss: 0.9468 - val_accuracy: 0.9317
Epoch 4/20
1500/1500 [=====] - 129s 86ms/step - loss: 0.2890 - accuracy: 0.9670 - val_loss: 0.9602 - val_accuracy: 0.9273
Epoch 5/20
1500/1500 [=====] - 122s 81ms/step - loss: 0.1933 - accuracy: 0.9745 - val_loss: 0.9956 - val_accuracy: 0.9303
Epoch 6/20
1500/1500 [=====] - 121s 81ms/step - loss: 0.1351 - accuracy: 0.9797 - val_loss: 1.0543 - val_accuracy: 0.9247
Epoch 7/20
1500/1500 [=====] - 129s 86ms/step - loss: 0.0859 - accuracy: 0.9850 - val_loss: 1.2734 - val_accuracy: 0.9250
Epoch 8/20
1500/1500 [=====] - 121s 80ms/step - loss: 0.0631 - accuracy: 0.9870 - val_loss: 1.1010 - val_accuracy: 0.9267
Epoch 9/20
1500/1500 [=====] - 121s 80ms/step - loss: 0.0305 - accuracy: 0.9932 - val_loss: 0.9703 - val_accuracy: 0.9340
Epoch 10/20
1500/1500 [=====] - 128s 85ms/step - loss: 0.0209 - accuracy: 0.9943 - val_loss: 1.0629 - val_accuracy: 0.9303
Epoch 11/20
1500/1500 [=====] - 121s 81ms/step - loss: 0.0107 - accuracy: 0.9969 - val_loss: 0.9757 - val_accuracy: 0.9353
Epoch 12/20
1500/1500 [=====] - 128s 85ms/step - loss: 0.0033 - accuracy: 0.9988 - val_loss: 0.9612 - val_accuracy: 0.9343
Epoch 13/20
1500/1500 [=====] - 129s 86ms/step - loss: 8.8030e-04 - accuracy: 0.9997 - val_loss: 0.9596 - val_accuracy: 0.9347
Epoch 14/20
1500/1500 [=====] - 129s 86ms/step - loss: 2.0187e-04 - accuracy: 1.0000 - val_loss: 0.9564 - val_accuracy: 0.9343
Epoch 15/20
1500/1500 [=====] - 121s 81ms/step - loss: 1.3301e-04 - accuracy: 1.0000 - val_loss: 0.9558 - val_accuracy: 0.9347
Epoch 16/20
1500/1500 [=====] - 121s 81ms/step - loss: 1.0825e-04 - accuracy: 1.0000 - val_loss: 0.9554 - val_accuracy: 0.9347
Epoch 17/20
1500/1500 [=====] - 121s 81ms/step - loss: 9.6528e-05 - accuracy: 1.0000 - val_loss: 0.9554 - val_accuracy: 0.9347
Epoch 18/20
1500/1500 [=====] - 129s 86ms/step - loss: 8.8897e-05 - accuracy: 1.0000 - val_loss: 0.9548 - val_accuracy: 0.9347
Epoch 19/20
1500/1500 [=====] - 121s 81ms/step - loss: 8.3106e-05 - accuracy: 1.0000 - val_loss: 0.9548 - val_accuracy: 0.9347
Epoch 20/20
1500/1500 [=====] - 121s 81ms/step - loss: 7.8616e-05 - accuracy: 1.0000 - val_loss: 0.9541 - val_accuracy: 0.9350
```

Almost nothing changes after around 10th epoch. Accuracy similar to other experiments, but has a very low validation loss,

Experiment 7: batch_size = 32, n_epochs = 20, learning rate = 0.001

```
Epoch 1/20
1500/1500 [=====] - 127s 84ms/step - loss: 11.2670 - accuracy: 0.9076 - val_loss: 10.9937 - val_accuracy: 0.910
Epoch 2/20
1500/1500 [=====] - 121s 80ms/step - loss: 5.9912 - accuracy: 0.9433 - val_loss: 8.8115 - val_accuracy: 0.9320
Epoch 3/20
1500/1500 [=====] - 121s 80ms/step - loss: 3.8204 - accuracy: 0.9577 - val_loss: 8.2725 - val_accuracy: 0.9340
Epoch 4/20
1500/1500 [=====] - 128s 85ms/step - loss: 2.4538 - accuracy: 0.9690 - val_loss: 9.2655 - val_accuracy: 0.9420
Epoch 5/20
1500/1500 [=====] - 128s 85ms/step - loss: 1.8802 - accuracy: 0.9721 - val_loss: 9.3162 - val_accuracy: 0.9303
Epoch 6/20
1500/1500 [=====] - 121s 81ms/step - loss: 1.3122 - accuracy: 0.9780 - val_loss: 8.7699 - val_accuracy: 0.9337
Epoch 7/20
1500/1500 [=====] - 121s 80ms/step - loss: 0.8002 - accuracy: 0.9843 - val_loss: 8.7504 - val_accuracy: 0.9303
Epoch 8/20
1500/1500 [=====] - 121s 81ms/step - loss: 0.7334 - accuracy: 0.9849 - val_loss: 8.1437 - val_accuracy: 0.9340
Epoch 9/20
1500/1500 [=====] - 121s 81ms/step - loss: 0.4286 - accuracy: 0.9895 - val_loss: 9.0134 - val_accuracy: 0.9310
Epoch 10/20
1500/1500 [=====] - 128s 85ms/step - loss: 0.3623 - accuracy: 0.9910 - val_loss: 8.7602 - val_accuracy: 0.9333
Epoch 11/20
1500/1500 [=====] - 121s 80ms/step - loss: 0.2338 - accuracy: 0.9930 - val_loss: 8.3824 - val_accuracy: 0.9360
Epoch 12/20
1500/1500 [=====] - 128s 85ms/step - loss: 0.2226 - accuracy: 0.9935 - val_loss: 7.9170 - val_accuracy: 0.9367
Epoch 13/20
1500/1500 [=====] - 128s 85ms/step - loss: 0.1868 - accuracy: 0.9942 - val_loss: 7.9453 - val_accuracy: 0.9370
Epoch 14/20
1500/1500 [=====] - 128s 85ms/step - loss: 0.1749 - accuracy: 0.9947 - val_loss: 8.2736 - val_accuracy: 0.9353
Epoch 15/20
1500/1500 [=====] - 128s 85ms/step - loss: 0.1329 - accuracy: 0.9954 - val_loss: 9.1686 - val_accuracy: 0.9377
Epoch 16/20
1500/1500 [=====] - 121s 80ms/step - loss: 0.0976 - accuracy: 0.9963 - val_loss: 8.8257 - val_accuracy: 0.9347
Epoch 17/20
1500/1500 [=====] - 128s 85ms/step - loss: 0.0902 - accuracy: 0.9965 - val_loss: 8.7585 - val_accuracy: 0.9320
Epoch 18/20
1500/1500 [=====] - 128s 86ms/step - loss: 0.0667 - accuracy: 0.9976 - val_loss: 8.1779 - val_accuracy: 0.9360
Epoch 19/20
1500/1500 [=====] - 128s 85ms/step - loss: 0.0824 - accuracy: 0.9971 - val_loss: 8.1109 - val_accuracy: 0.9380
Epoch 20/20
1500/1500 [=====] - 128s 85ms/step - loss: 0.0354 - accuracy: 0.9983 - val_loss: 8.5214 - val_accuracy: 0.9340
```

Lower batch size allowed the model to converge earlier (compared to same learning rate with higher batch size)

Experiment 8: batch_size = 16, n_epochs = 20, learning rate = 0.0001

```
Epoch 1/20
1500/1500 [=====] - 129s 86ms/step - loss: 1.5570 - accuracy: 0.9050 - val_loss: 1.0018 - val_accuracy: 0.9290
Epoch 2/20
1500/1500 [=====] - 121s 80ms/step - loss: 0.7170 - accuracy: 0.9435 - val_loss: 1.1405 - val_accuracy: 0.9340
Epoch 3/20
1500/1500 [=====] - 121s 80ms/step - loss: 0.4574 - accuracy: 0.9576 - val_loss: 0.9015 - val_accuracy: 0.9357
Epoch 4/20
1500/1500 [=====] - 120s 80ms/step - loss: 0.2953 - accuracy: 0.9662 - val_loss: 0.9504 - val_accuracy: 0.9363
Epoch 5/20
1500/1500 [=====] - 128s 85ms/step - loss: 0.1995 - accuracy: 0.9732 - val_loss: 0.9305 - val_accuracy: 0.9343
Epoch 6/20
1500/1500 [=====] - 128s 85ms/step - loss: 0.1355 - accuracy: 0.9793 - val_loss: 0.9727 - val_accuracy: 0.9347
Epoch 7/20
1500/1500 [=====] - 121s 81ms/step - loss: 0.0904 - accuracy: 0.9854 - val_loss: 0.9232 - val_accuracy: 0.9323
Epoch 8/20
1500/1500 [=====] - 122s 81ms/step - loss: 0.0621 - accuracy: 0.9877 - val_loss: 1.0025 - val_accuracy: 0.9317
Epoch 9/20
1500/1500 [=====] - 121s 81ms/step - loss: 0.0283 - accuracy: 0.9927 - val_loss: 1.1665 - val_accuracy: 0.9300
Epoch 10/20
1500/1500 [=====] - 129s 86ms/step - loss: 0.0218 - accuracy: 0.9945 - val_loss: 0.9624 - val_accuracy: 0.9313
Epoch 11/20
1500/1500 [=====] - 128s 85ms/step - loss: 0.0115 - accuracy: 0.9961 - val_loss: 0.9533 - val_accuracy: 0.9327
Epoch 12/20
1500/1500 [=====] - 121s 81ms/step - loss: 0.0044 - accuracy: 0.9985 - val_loss: 0.9235 - val_accuracy: 0.9297
Epoch 13/20
1500/1500 [=====] - 121s 80ms/step - loss: 0.0019 - accuracy: 0.9993 - val_loss: 0.9340 - val_accuracy: 0.9347
Epoch 14/20
1500/1500 [=====] - 121s 80ms/step - loss: 6.3376e-04 - accuracy: 0.9999 - val_loss: 0.9191 - val_accuracy: 0.9317
Epoch 15/20
1500/1500 [=====] - 121s 81ms/step - loss: 1.5717e-04 - accuracy: 1.0000 - val_loss: 0.9166 - val_accuracy: 0.9307
Epoch 16/20
1500/1500 [=====] - 121s 80ms/step - loss: 1.0792e-04 - accuracy: 1.0000 - val_loss: 0.9164 - val_accuracy: 0.9313
Epoch 17/20
1500/1500 [=====] - 128s 85ms/step - loss: 9.4364e-05 - accuracy: 1.0000 - val_loss: 0.9143 - val_accuracy: 0.9307
Epoch 18/20
1500/1500 [=====] - 122s 81ms/step - loss: 8.6359e-05 - accuracy: 1.0000 - val_loss: 0.9149 - val_accuracy: 0.9303
Epoch 19/20
1500/1500 [=====] - 122s 81ms/step - loss: 7.9691e-05 - accuracy: 1.0000 - val_loss: 0.9143 - val_accuracy: 0.9307
Epoch 20/20
1500/1500 [=====] - 122s 81ms/step - loss: 7.4844e-05 - accuracy: 1.0000 - val_loss: 0.9150 - val_accuracy: 0.9307
```

Unfortunately, Google Colab banned me from using gpu at this point, they probably thought I was crypto mining with their gpu's – since I was experimenting for around 8 hours. Using cpu is not an option as a single epoch takes around 3 hours. I am not even sure if those experiments had a high importance (I may have used better hyperparameter options) anyways, since values are all very close to each other (but achieving low loss value with higher accuracy is good for test samples), but it was fun at the least. I will not be continuing experimenting since they may also ban me from my gmail account as well, so some of the hyperparameters combinations are not tested (I also wanted to try Adams vs SGD). Therefore, I will be using my own gmail account to finish the code, and use the optimal values found so far.

Validation Accuracy (Lowest/Highest/Converged):

Learning Rate\Batch Size	8	16	32	64
0.01			0.9217 – 0.9420 – 0.9400	0.9177 – 0.9410 – 0.94
0.001	0.9053 - 0.9397 – 93.70		0.9100 – 0.9420 – 0.9340	0.9100 – 0.9407 – 0.9380
0.0001		0.9290 – 0.9363 – 0.9307		0.9037 – 0.9377 – 0.9347

Validation Loss (for the given lowest, highest and converged accuracy):

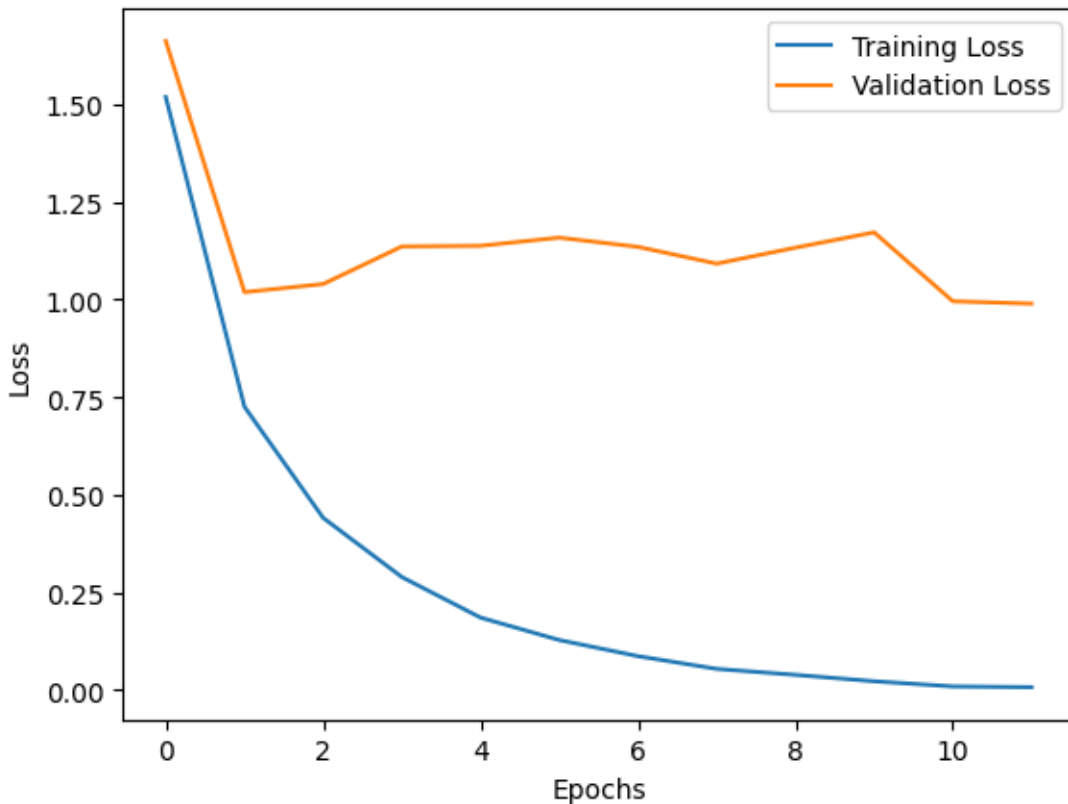
Learning Rate\Batch Size	8	16	32	64
0.01			94 – 74 – 80	113.1 – 81.2 - 75
0.001	13.9 – 7.8 – 8.5		10.99 – 9 – 8.5	13.83 – 7.49 – 8.5
0.0001		1 – 0.95 – 0.91		1.53 – 0.98 – 0.95

- Using the values found from the experiments, and the following article: [The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset - ScienceDirect](#) , I will be using learning rate of 0.0001 and batch size of 64, and will train for 12 epochs (that's where convergence happened for these values). Even though there are hyperparameters with higher accuracies, there is a very high difference between validation losses (yet, they still might be the better option considering higher accuracy is the main goal).

Results:

```
Epoch 1/12  
1500/1500 [=====] - 136s 84ms/step - loss: 1.5187 - accuracy: 0.9051 - val_loss: 1.6624 - val_accuracy: 0.9133  
Epoch 2/12  
1500/1500 [=====] - 121s 81ms/step - loss: 0.7254 - accuracy: 0.9415 - val_loss: 1.0189 - val_accuracy: 0.9277  
Epoch 3/12  
1500/1500 [=====] - 128s 85ms/step - loss: 0.4408 - accuracy: 0.9574 - val_loss: 1.0397 - val_accuracy: 0.9327  
Epoch 4/12  
1500/1500 [=====] - 129s 86ms/step - loss: 0.2892 - accuracy: 0.9668 - val_loss: 1.1358 - val_accuracy: 0.9260  
Epoch 5/12  
1500/1500 [=====] - 128s 85ms/step - loss: 0.1854 - accuracy: 0.9752 - val_loss: 1.1376 - val_accuracy: 0.9247  
Epoch 6/12  
1500/1500 [=====] - 121s 81ms/step - loss: 0.1280 - accuracy: 0.9799 - val_loss: 1.1585 - val_accuracy: 0.9273  
Epoch 7/12  
1500/1500 [=====] - 121s 81ms/step - loss: 0.0866 - accuracy: 0.9846 - val_loss: 1.1348 - val_accuracy: 0.9250  
Epoch 8/12  
1500/1500 [=====] - 128s 85ms/step - loss: 0.0541 - accuracy: 0.9890 - val_loss: 1.0919 - val_accuracy: 0.9297  
Epoch 9/12  
1500/1500 [=====] - 121s 81ms/step - loss: 0.0388 - accuracy: 0.9908 - val_loss: 1.1327 - val_accuracy: 0.9267  
Epoch 10/12  
1500/1500 [=====] - 121s 80ms/step - loss: 0.0224 - accuracy: 0.9935 - val_loss: 1.1721 - val_accuracy: 0.9250  
Epoch 11/12  
1500/1500 [=====] - 129s 86ms/step - loss: 0.0092 - accuracy: 0.9969 - val_loss: 0.9955 - val_accuracy: 0.9310  
Epoch 12/12  
1500/1500 [=====] - 121s 81ms/step - loss: 0.0072 - accuracy: 0.9973 - val_loss: 0.9895 - val_accuracy: 0.9320
```

above is the train and val values of the final model, and below is the loss graph of this model



- Using this model, 0.933 test accuracy is achieved.
- Even though this is not the highest accuracy that can be achieved, still it performed better than most of the validation accuracy that had different hyperparameters. In addition, since the loss is way lower than other hyperparameters, test outputs are more consistent, accuracy does not change that much depending on the test data.