Name: Aya Abdelmoniem ID: 20221462478

Name: Doaa Samir ID: 2103114

REPORT

what is ResNet?

ResNet stands for residual network and it contains more than 150 layers and can overcome the problem of vanishing gradients.

As CNNs grow deeper, vanishing gradient tend to occur which negatively impact network performance.

Vanishing gradient problem occurs when the gradient is back-propagated to earlier layers which results in a very small gradient.

ResNets include "skip connection" feature which enables training of multiple deep layers, 152 layers without vanishing gradient issues.

what is Vgg?

VGG stands for Visual Geometry Group and consists of blocks, where each block is composed of 2D Convolution and Max Pooling layers. It comes in two models — VGG16 and VGG19 — with 16 and 19 layers.

Vgg used small filters because of fewer parameters and stack more of them instead of having large filters.

It has 16 in this case for Vgg16, and then 19 for Vgg19, it's a very similar architecture, but with a few more convolution layers in there.

As the number of layers increases in CNN, the ability of the model to fit more complex functions also increases. Hence, more layers promise better performance.

Difference between Vgg and ResNet?

Architecture: VGG has a simpler architecture compared to ResNet. VGG uses a stack of convolutional layers with small 3x3 filters, while ResNet uses a deeper network with shortcut connections that allow informationto pass directly across layers without going through all the intermediate layers. Therefore ResNet can be trained to a much deeper level without suffering from the vanishing gradient problem that limits traditional CNN.

Performance: ResNet has outperform VGG , especially in image classification with extremely deep architectures.

Memory usage: ResNet requires more memory than VGG due to its deeper architecture and the additional shortcut connections.

Training time: VGG can be trained faster than ResNet.

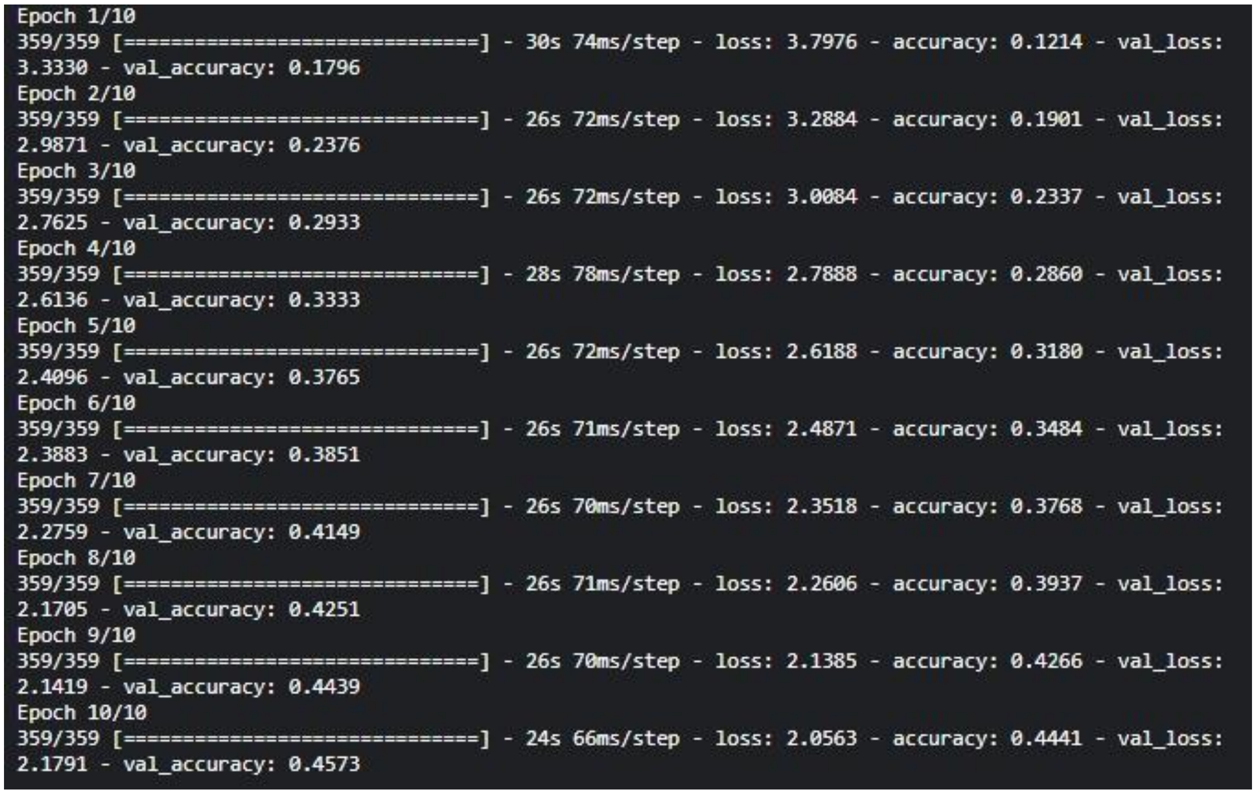
ResNet is a deeper, more complex architecture that can achieve better results but requires more memory and training time. While VGG has a simpler architecture that is faster to train but may not perform as well on extremely deep architectures.

How did we access and load the data?

First we did load to the data through image from directory, also we made the validation as 0.1 and the batch size as 32 and we splitted it into training and validation.

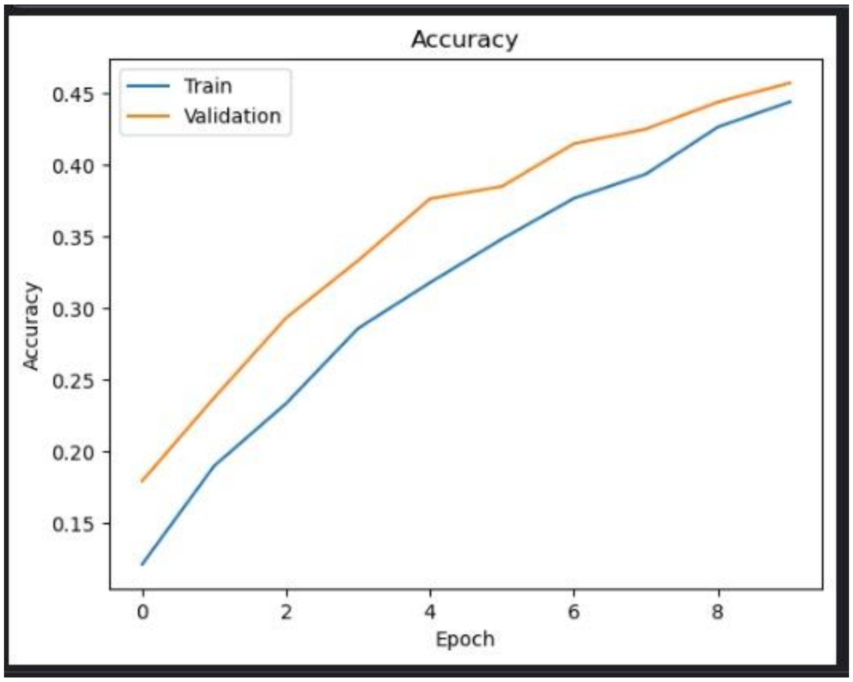
In the test we also did the batch size as 3712, so we have one batch having 3712 file and then we used next() function in order to bring the batch in the test data, and then we used [0] to access images and [1] to access labels.

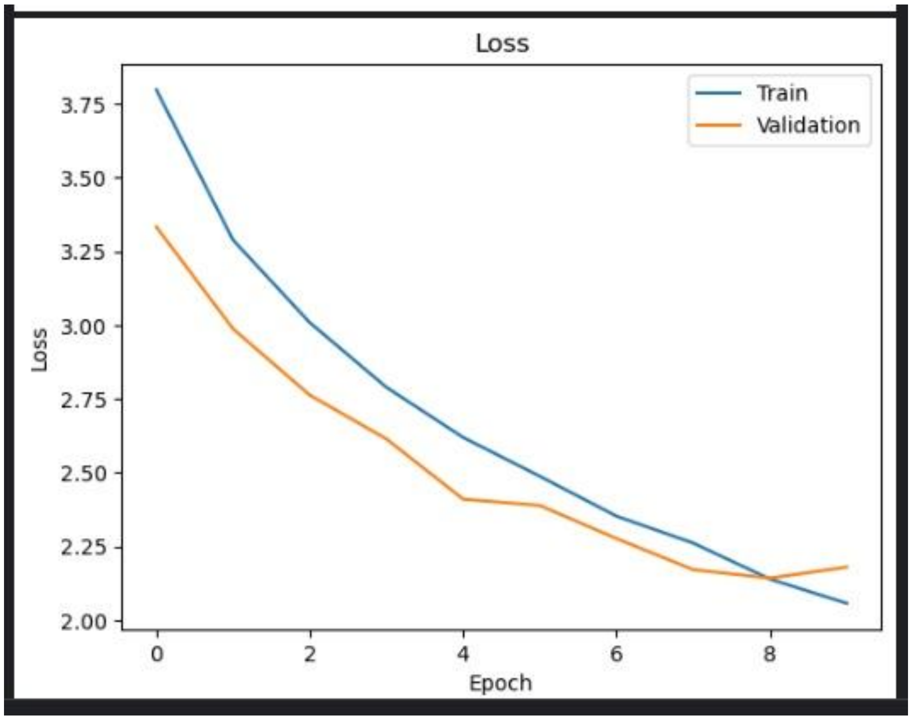
Output of our simple model



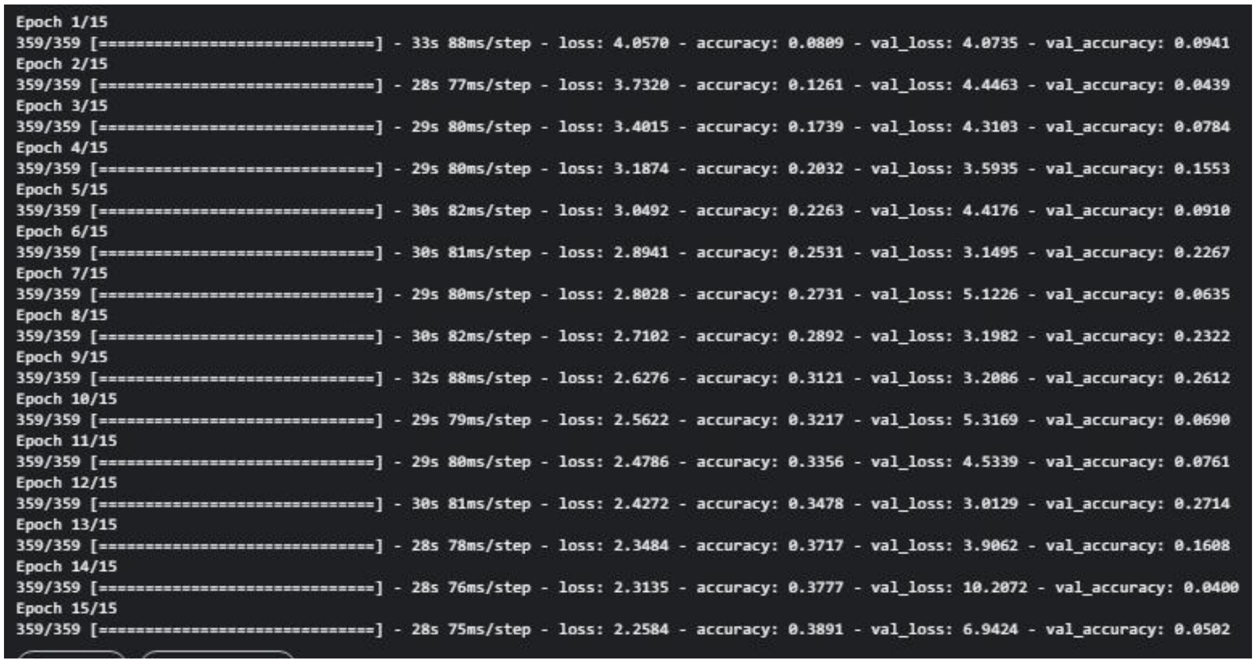
```
Epoch 1/10
359/359 [==============================] - 30s 74ms/step - loss: 3.7976 - accuracy: 0.1214 - val_loss:
3.3330 - val_accuracy: 0.1796
Epoch 2/10
359/359 [==============================] - 26s 72ms/step - loss: 3.2884 - accuracy: 0.1901 - val_loss:
2.9871 - val_accuracy: 0.2376
Epoch 3/10
359/359 [==============================] - 26s 72ms/step - loss: 3.0084 - accuracy: 0.2337 - val_loss:
2.7625 - val_accuracy: 0.2933
Epoch 4/10
359/359 [==============================] - 28s 78ms/step - loss: 2.7888 - accuracy: 0.2860 - val_loss:
2.6136 - val_accuracy: 0.3333
Epoch 5/10
359/359 [==============================] - 26s 72ms/step - loss: 2.6188 - accuracy: 0.3180 - val_loss:
2.4096 - val_accuracy: 0.3765
Epoch 6/10
359/359 [==============================] - 26s 71ms/step - loss: 2.4871 - accuracy: 0.3484 - val_loss:
2.3883 - val_accuracy: 0.3851
Epoch 7/10
359/359 [==============================] - 26s 70ms/step - loss: 2.3518 - accuracy: 0.3768 - val_loss:
2.2759 - val_accuracy: 0.4149
Epoch 8/10
359/359 [==============================] - 26s 71ms/step - loss: 2.2606 - accuracy: 0.3937 - val_loss:
2.1705 - val_accuracy: 0.4251
Epoch 9/10
359/359 [==============================] - 26s 70ms/step - loss: 2.1385 - accuracy: 0.4266 - val_loss:
2.1419 - val_accuracy: 0.4439
Epoch 10/10
359/359 [==============================] - 24s 66ms/step - loss: 2.0563 - accuracy: 0.4441 - val_loss:
2.1791 - val_accuracy: 0.4573
```

Plots



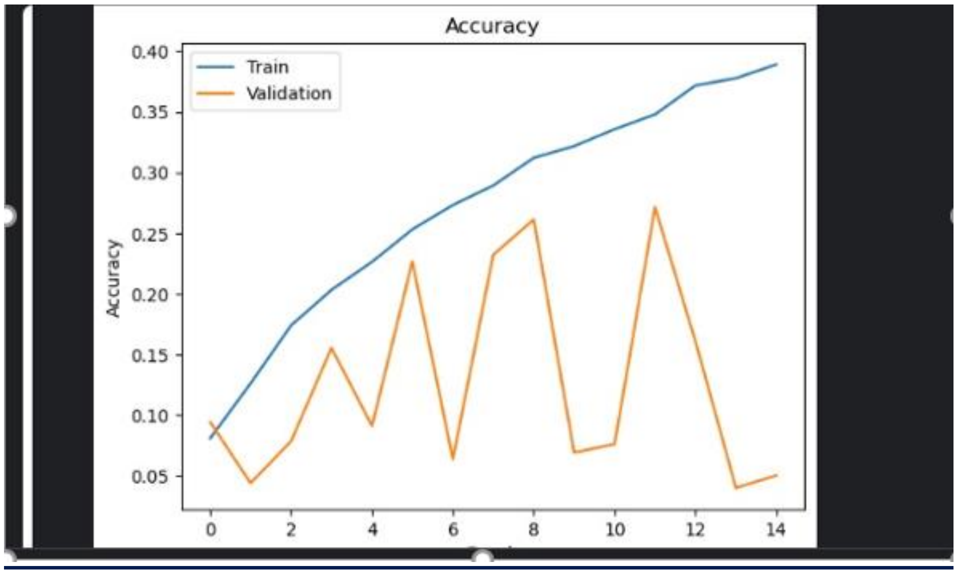


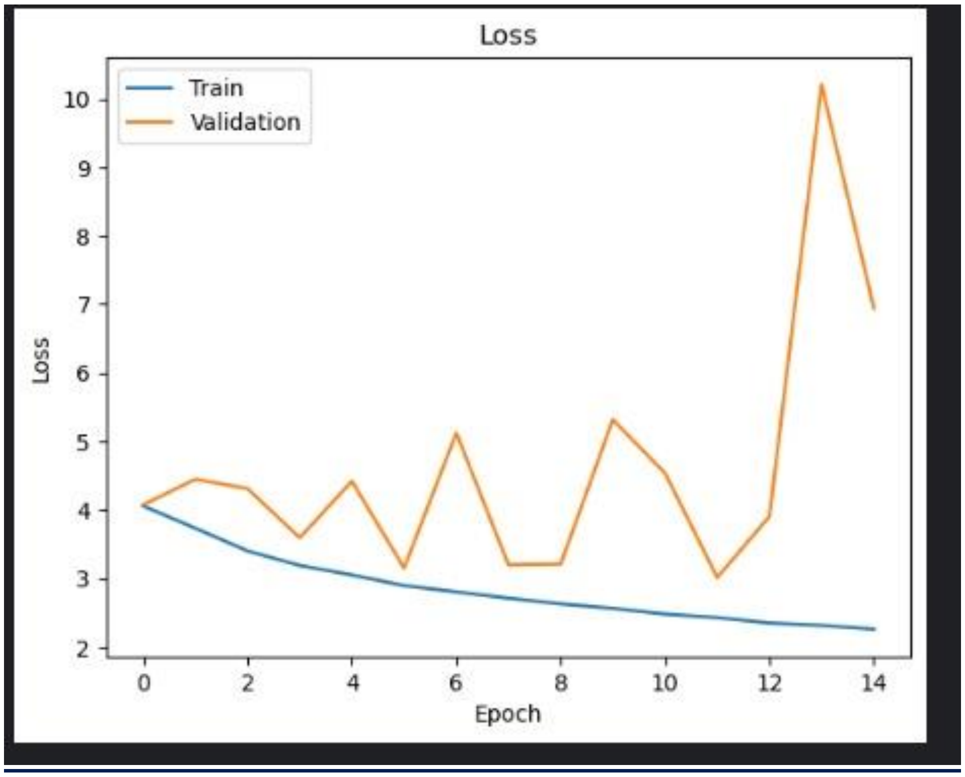
This is how we got the accuracy and the loss after we added dropout and some layers, so you can see that the accuracy is increasing while the loss is decreasing.



```
Epoch 1/15
359/359 [==============================] - 33s 88ms/step - loss: 4.0570 - accuracy: 0.0809 - val_loss: 4.0735 - val_accuracy: 0.0941
Epoch 2/15
359/359 [==============================] - 28s 77ms/step - loss: 3.7320 - accuracy: 0.1261 - val_loss: 4.4463 - val_accuracy: 0.0439
Epoch 3/15
359/359 [==============================] - 29s 80ms/step - loss: 3.4015 - accuracy: 0.1739 - val_loss: 4.3103 - val_accuracy: 0.0784
Epoch 4/15
359/359 [==============================] - 29s 80ms/step - loss: 3.1874 - accuracy: 0.2032 - val_loss: 3.5935 - val_accuracy: 0.1553
Epoch 5/15
359/359 [==============================] - 30s 82ms/step - loss: 3.0492 - accuracy: 0.2263 - val_loss: 4.4176 - val_accuracy: 0.0910
Epoch 6/15
359/359 [==============================] - 30s 81ms/step - loss: 2.8941 - accuracy: 0.2531 - val_loss: 3.1495 - val_accuracy: 0.2267
Epoch 7/15
359/359 [==============================] - 29s 80ms/step - loss: 2.8028 - accuracy: 0.2731 - val_loss: 5.1226 - val_accuracy: 0.0635
Epoch 8/15
359/359 [==============================] - 30s 82ms/step - loss: 2.7102 - accuracy: 0.2892 - val_loss: 3.1982 - val_accuracy: 0.2322
Epoch 9/15
359/359 [==============================] - 32s 88ms/step - loss: 2.6276 - accuracy: 0.3121 - val_loss: 3.2086 - val_accuracy: 0.2612
Epoch 10/15
359/359 [==============================] - 29s 79ms/step - loss: 2.5622 - accuracy: 0.3217 - val_loss: 5.3169 - val_accuracy: 0.0690
Epoch 11/15
359/359 [==============================] - 29s 80ms/step - loss: 2.4786 - accuracy: 0.3356 - val_loss: 4.5339 - val_accuracy: 0.0761
Epoch 12/15
359/359 [==============================] - 30s 81ms/step - loss: 2.4272 - accuracy: 0.3478 - val_loss: 3.0129 - val_accuracy: 0.2714
Epoch 13/15
359/359 [==============================] - 28s 78ms/step - loss: 2.3484 - accuracy: 0.3717 - val_loss: 3.9062 - val_accuracy: 0.1608
Epoch 14/15
359/359 [==============================] - 28s 76ms/step - loss: 2.3135 - accuracy: 0.3777 - val_loss: 10.2072 - val_accuracy: 0.0400
Epoch 15/15
359/359 [==============================] - 28s 75ms/step - loss: 2.2584 - accuracy: 0.3891 - val_loss: 6.9424 - val_accuracy: 0.0502
```

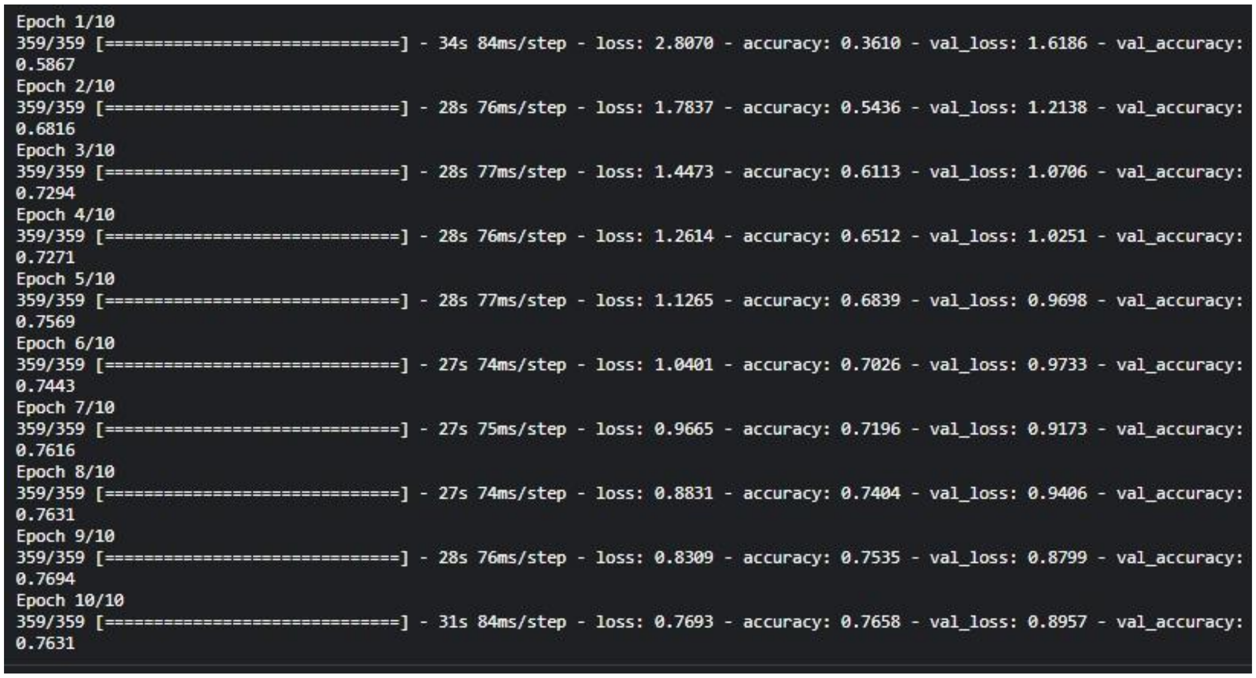
Plots





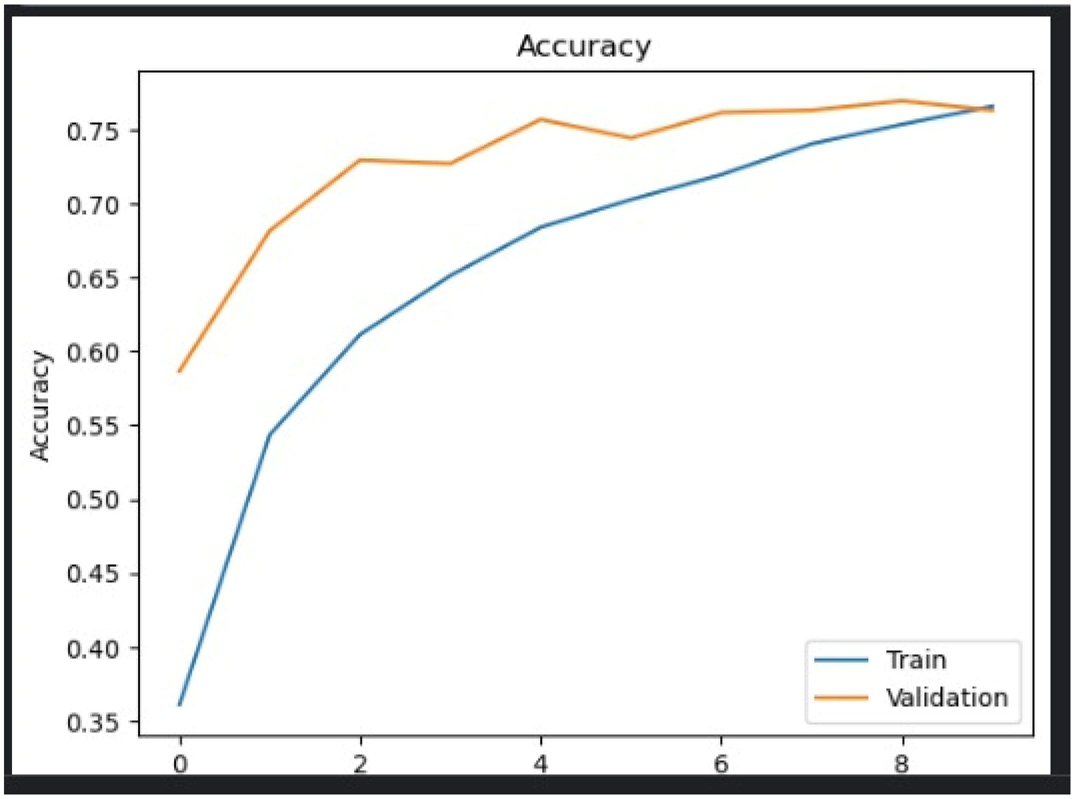
This the result after we increased the layers and the number of filters as you can see, in the plot of the accuracy the train is increasing while the validation keeps on decreasing and increasing at the same time. In the plot of the loss the validation keeps on decreasing and increasing while the train in decreasing.

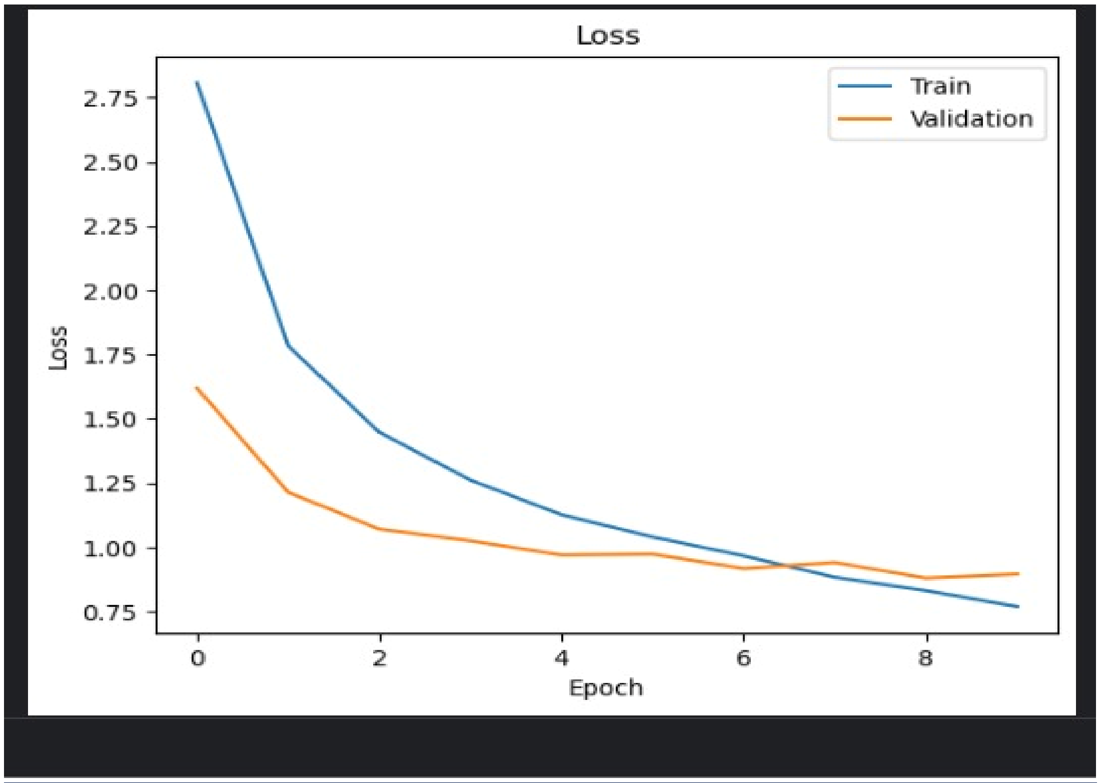
Output of ResNet50



```
Epoch 1/10
359/359 [==============================] - 34s 84ms/step - loss: 2.8070 - accuracy: 0.3610 - val_loss: 1.6186 - val_accuracy:
0.5867
Epoch 2/10
359/359 [==============================] - 28s 76ms/step - loss: 1.7837 - accuracy: 0.5436 - val_loss: 1.2138 - val_accuracy:
0.6816
Epoch 3/10
359/359 [==============================] - 28s 77ms/step - loss: 1.4473 - accuracy: 0.6113 - val_loss: 1.0706 - val_accuracy:
0.7294
Epoch 4/10
359/359 [==============================] - 28s 76ms/step - loss: 1.2614 - accuracy: 0.6512 - val_loss: 1.0251 - val_accuracy:
0.7271
Epoch 5/10
359/359 [==============================] - 28s 77ms/step - loss: 1.1265 - accuracy: 0.6839 - val_loss: 0.9698 - val_accuracy:
0.7569
Epoch 6/10
359/359 [==============================] - 27s 74ms/step - loss: 1.0401 - accuracy: 0.7026 - val_loss: 0.9733 - val_accuracy:
0.7443
Epoch 7/10
359/359 [==============================] - 27s 75ms/step - loss: 0.9665 - accuracy: 0.7196 - val_loss: 0.9173 - val_accuracy:
0.7616
Epoch 8/10
359/359 [==============================] - 27s 74ms/step - loss: 0.8831 - accuracy: 0.7404 - val_loss: 0.9406 - val_accuracy:
0.7631
Epoch 9/10
359/359 [==============================] - 28s 76ms/step - loss: 0.8309 - accuracy: 0.7535 - val_loss: 0.8799 - val_accuracy:
0.7694
Epoch 10/10
359/359 [==============================] - 31s 84ms/step - loss: 0.7693 - accuracy: 0.7658 - val_loss: 0.8957 - val_accuracy:
0.7631
```

Plots

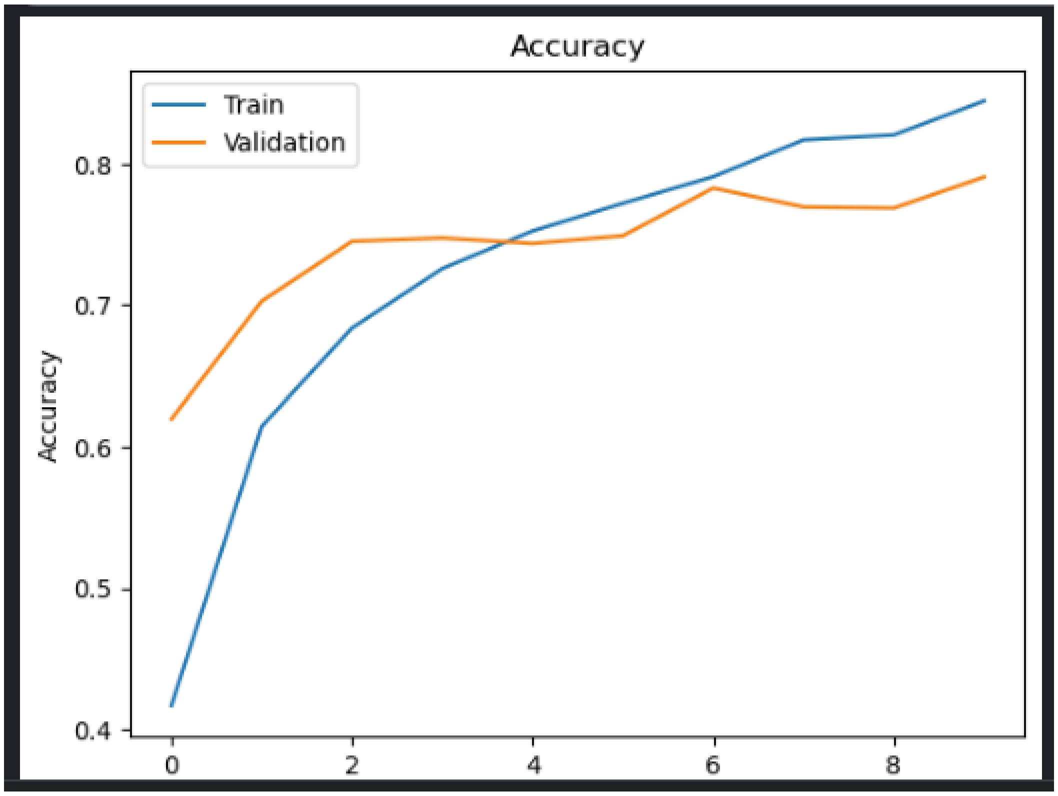

Accuracy
Accuracy
0.35
0.40
0.45
0.50
0.55
0.60
0.65
0.70
0.75
0
2
4
6
8
Train
Validation


Loss
Loss
0.75
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
0
2
4
6
8
Epoch
Train
Validation

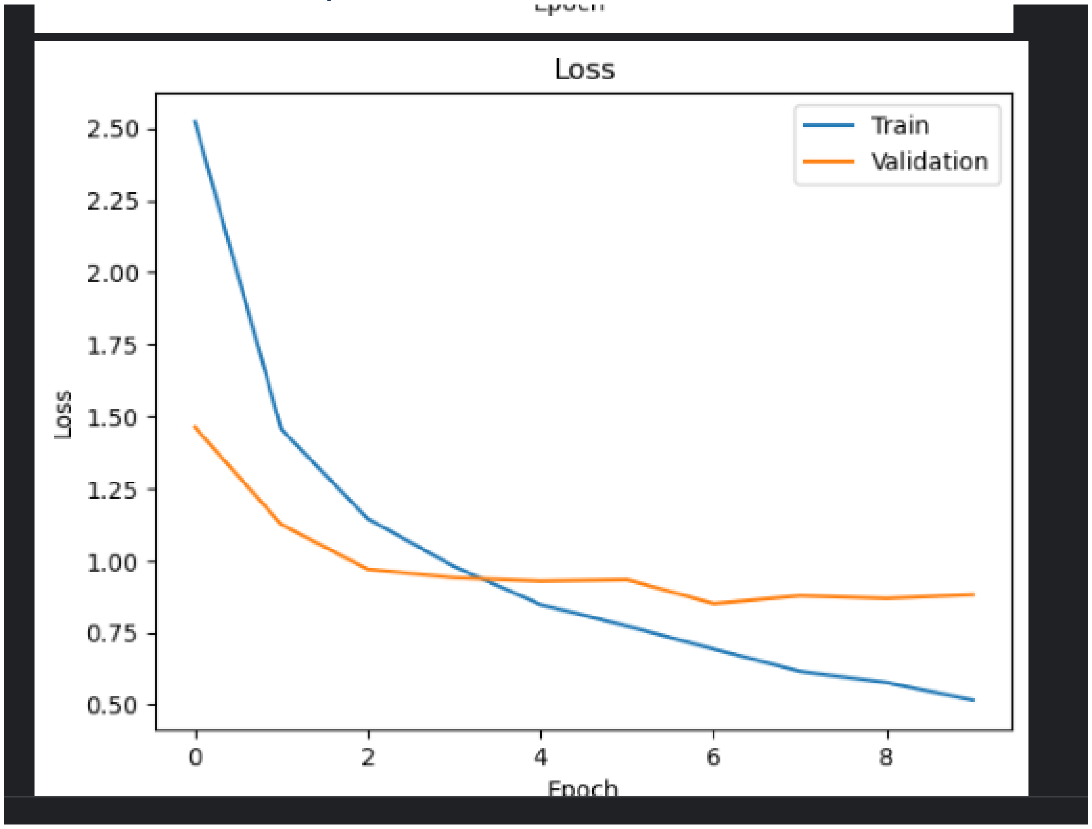
When we added these dropout, layers, the model gives us high accuracy.

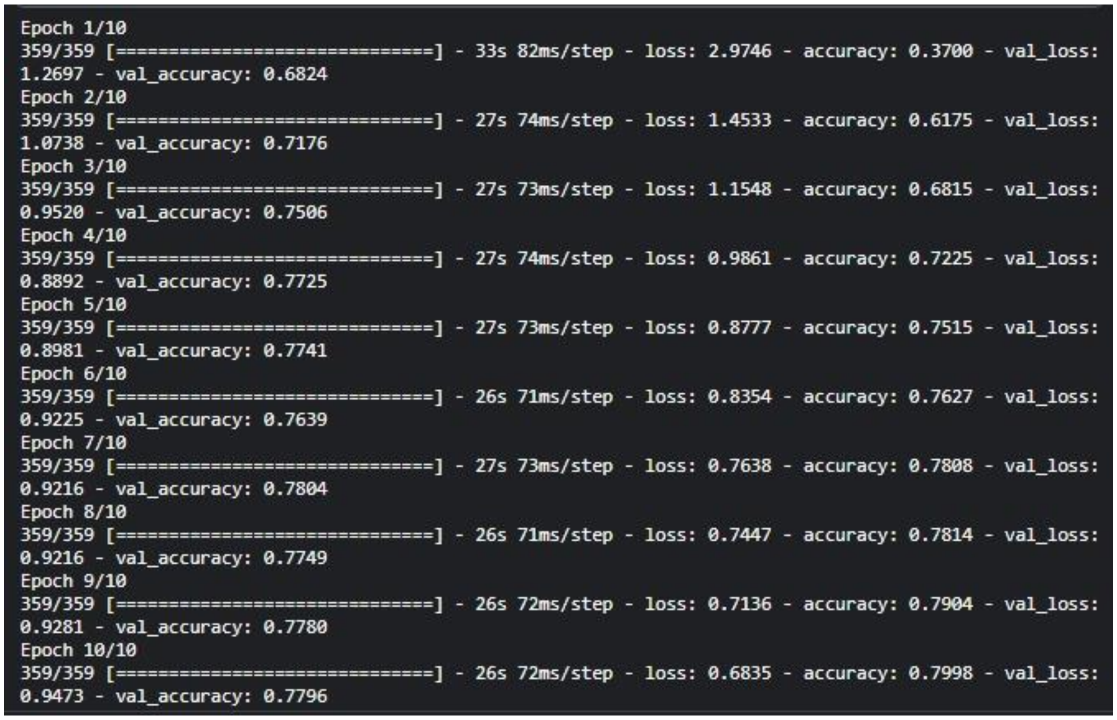
After we added some layers after the cnn and the dropout, the accuracy got high and less overfitting was there.

<u>acc after add drop and more nn loss:</u>



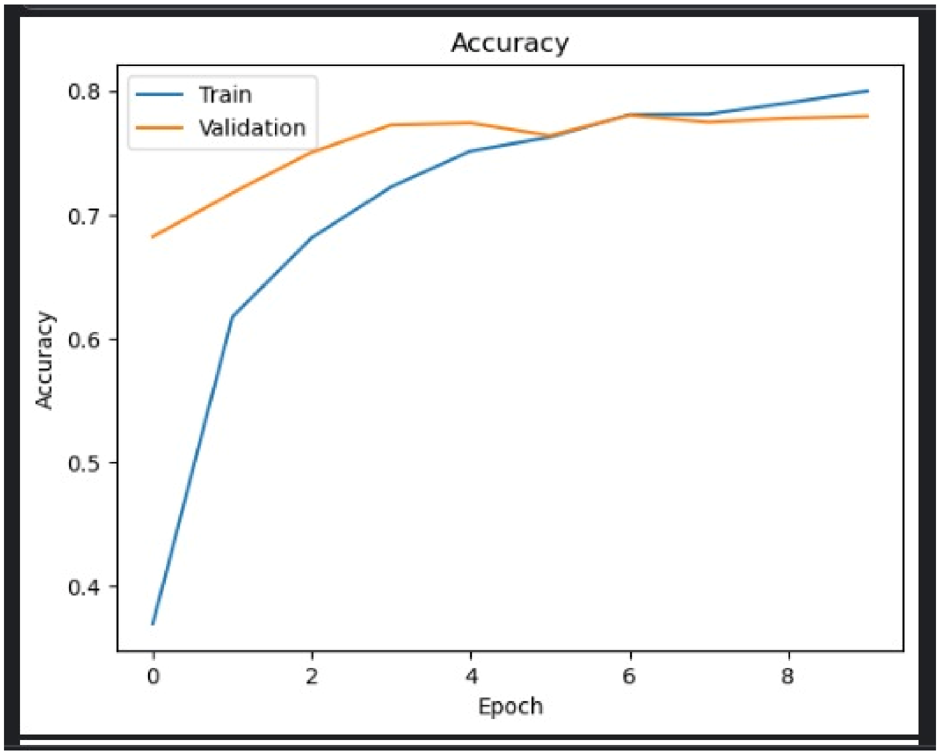
acc after add drop and more nn acc:

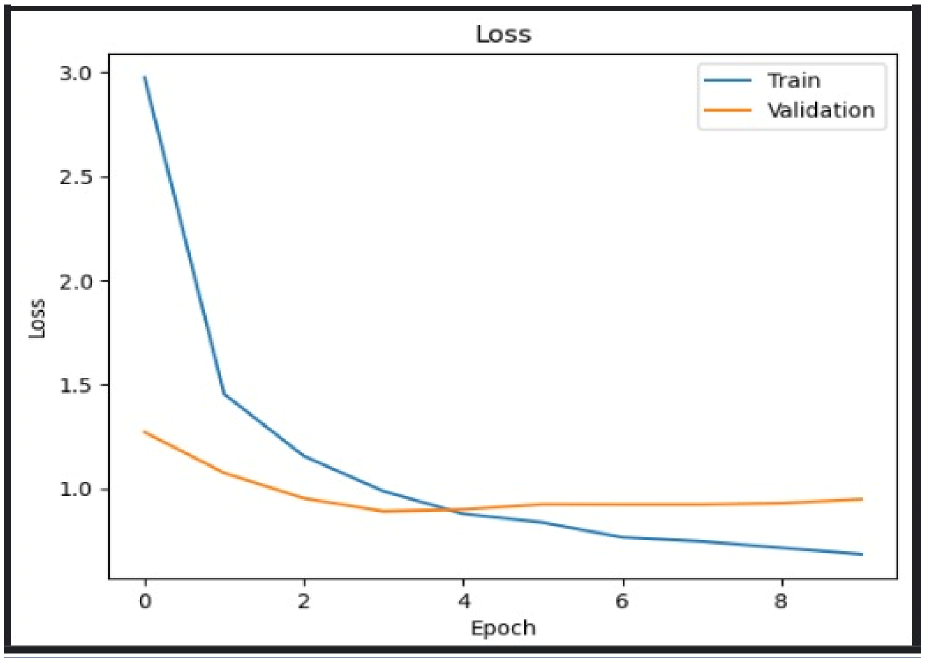




```
Epoch 1/10
359/359 [==============================] - 33s 82ms/step - loss: 2.9746 - accuracy: 0.3700 - val_loss:
1.2697 - val_accuracy: 0.6824
Epoch 2/10
359/359 [==============================] - 27s 74ms/step - loss: 1.4533 - accuracy: 0.6175 - val_loss:
1.0738 - val_accuracy: 0.7176
Epoch 3/10
359/359 [==============================] - 27s 73ms/step - loss: 1.1548 - accuracy: 0.6815 - val_loss:
0.9520 - val_accuracy: 0.7506
Epoch 4/10
359/359 [==============================] - 27s 74ms/step - loss: 0.9861 - accuracy: 0.7225 - val_loss:
0.8892 - val_accuracy: 0.7725
Epoch 5/10
359/359 [==============================] - 27s 73ms/step - loss: 0.8777 - accuracy: 0.7515 - val_loss:
0.8981 - val_accuracy: 0.7741
Epoch 6/10
359/359 [==============================] - 26s 71ms/step - loss: 0.8354 - accuracy: 0.7627 - val_loss:
0.9225 - val_accuracy: 0.7639
Epoch 7/10
359/359 [==============================] - 27s 73ms/step - loss: 0.7638 - accuracy: 0.7808 - val_loss:
0.9216 - val_accuracy: 0.7804
Epoch 8/10
359/359 [==============================] - 26s 71ms/step - loss: 0.7447 - accuracy: 0.7814 - val_loss:
0.9216 - val_accuracy: 0.7749
Epoch 9/10
359/359 [==============================] - 26s 72ms/step - loss: 0.7136 - accuracy: 0.7904 - val_loss:
0.9281 - val_accuracy: 0.7780
Epoch 10/10
359/359 [==============================] - 26s 72ms/step - loss: 0.6835 - accuracy: 0.7998 - val_loss:
0.9473 - val_accuracy: 0.7796
```

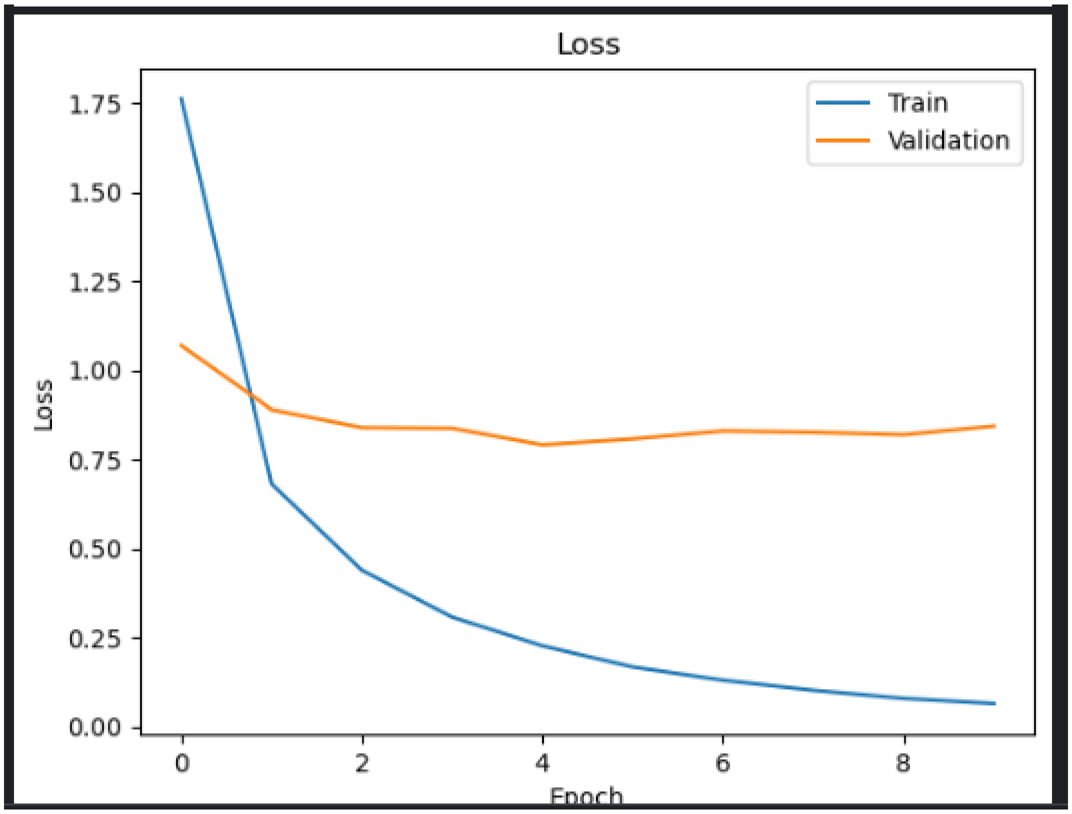
Plots


Accuracy
Train
Validation
Accuracy
0.4
0.5
0.6
0.7
0.8
0
2
4
6
8
Epoch

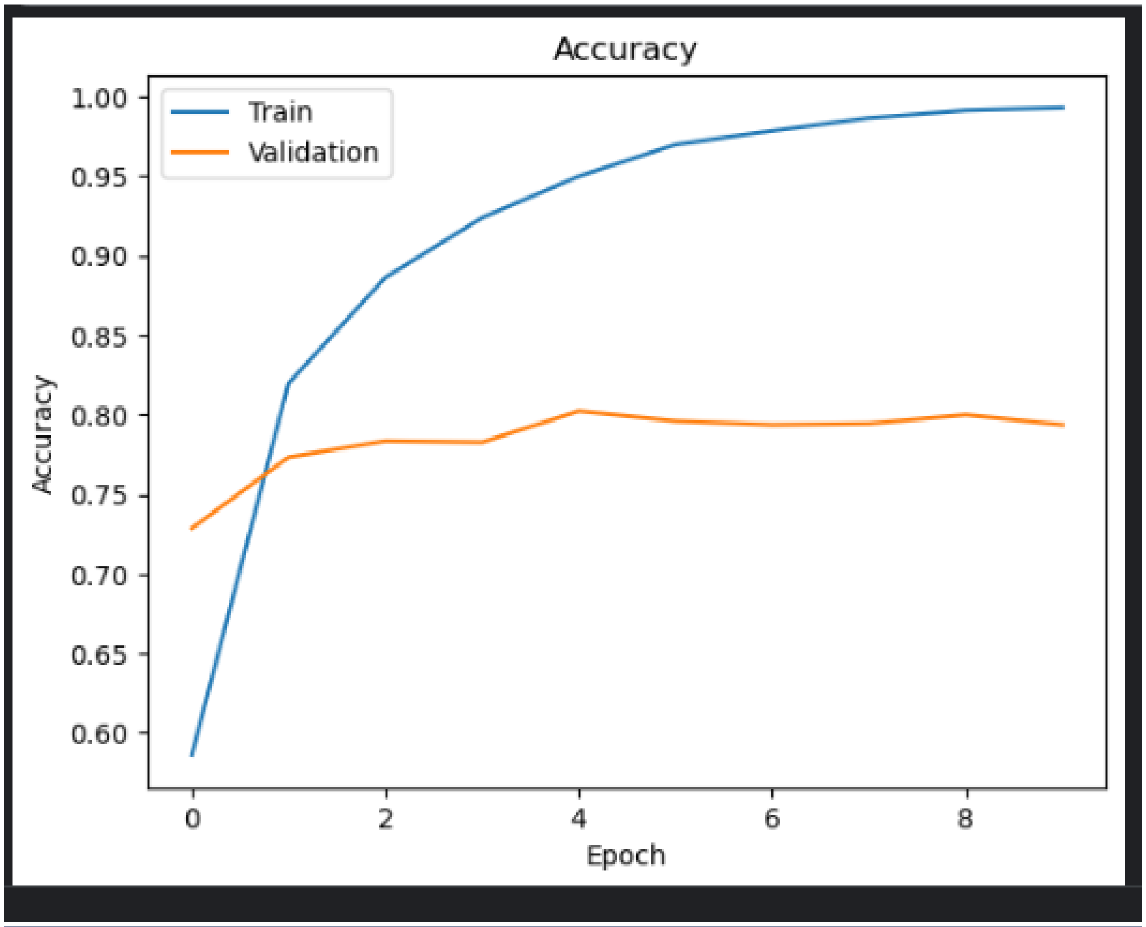

Loss
Train
Validation
Loss
1.0
1.5
2.0
2.5
3.0
0
2
4
6
8
Epoch

There was overfitting here since we didn't add the dropout and layers, the accuracy was ok but still not the best either.

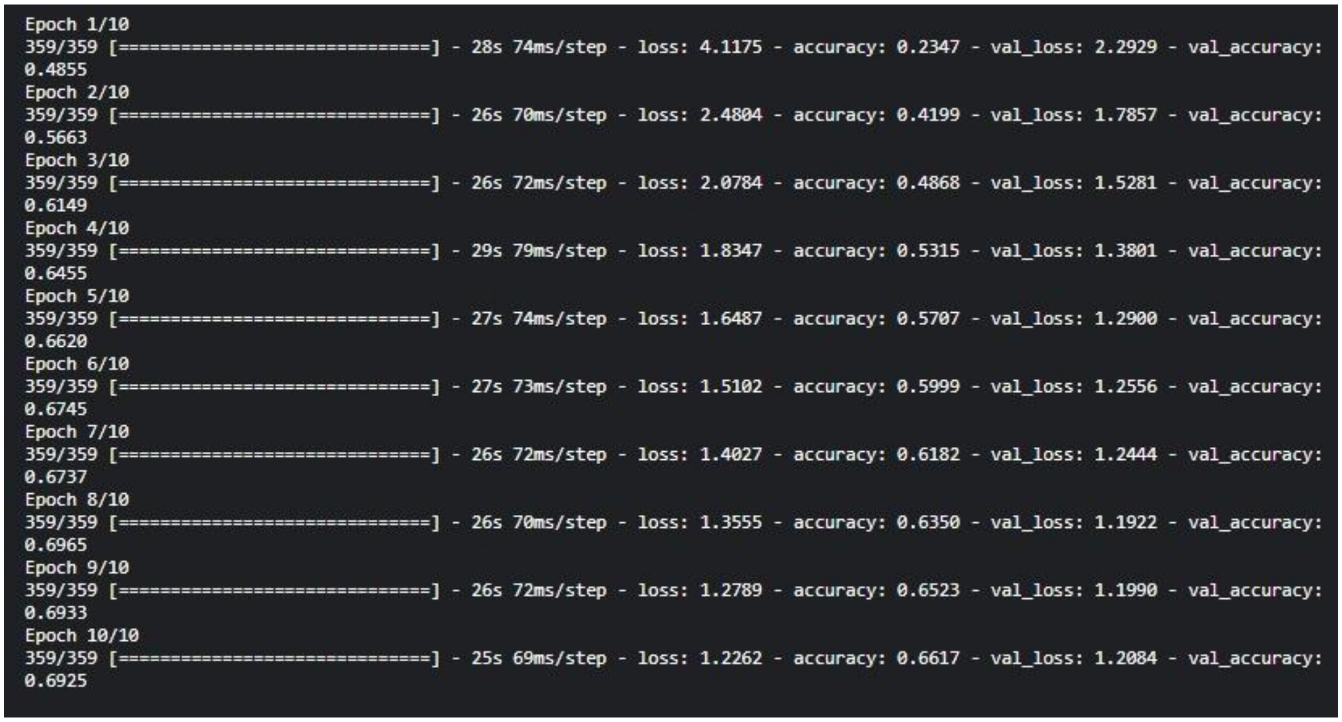
resnet without drop out loss:



resnet without drop out acc:

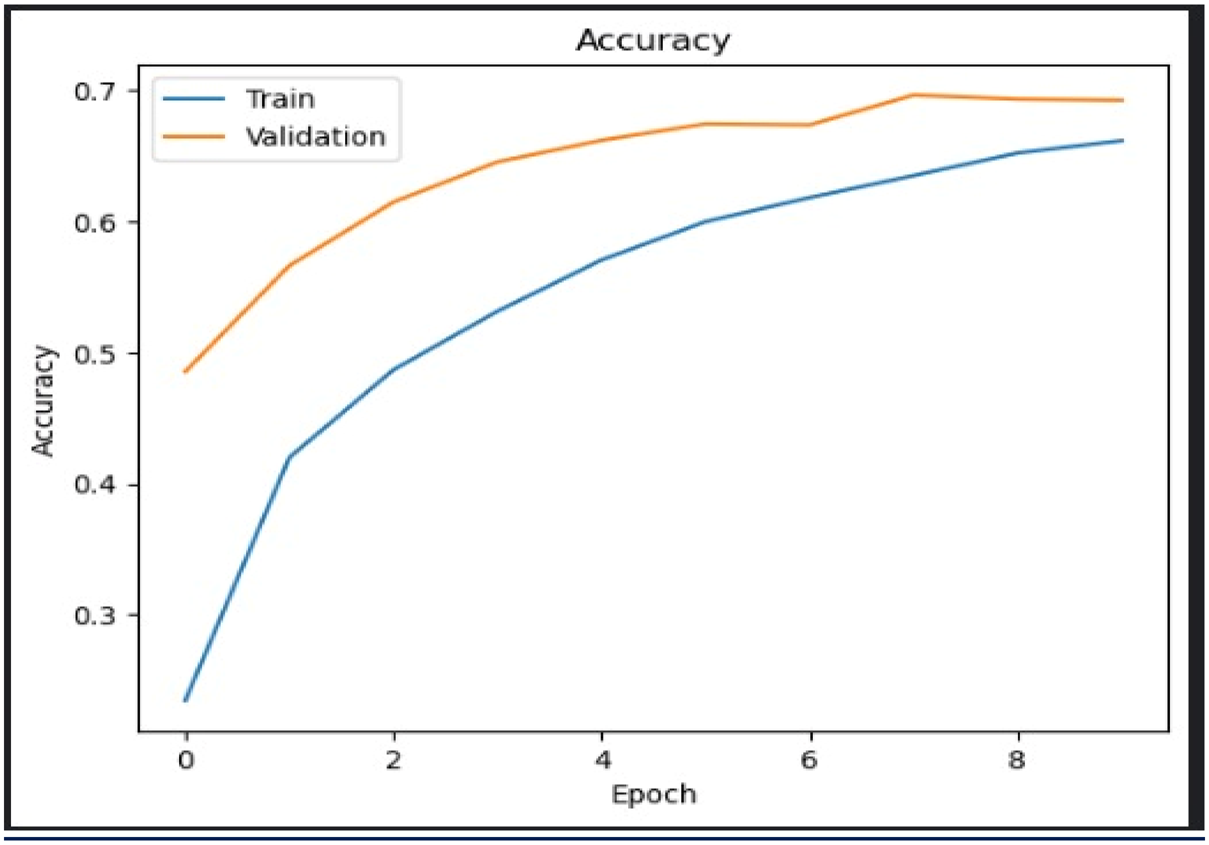


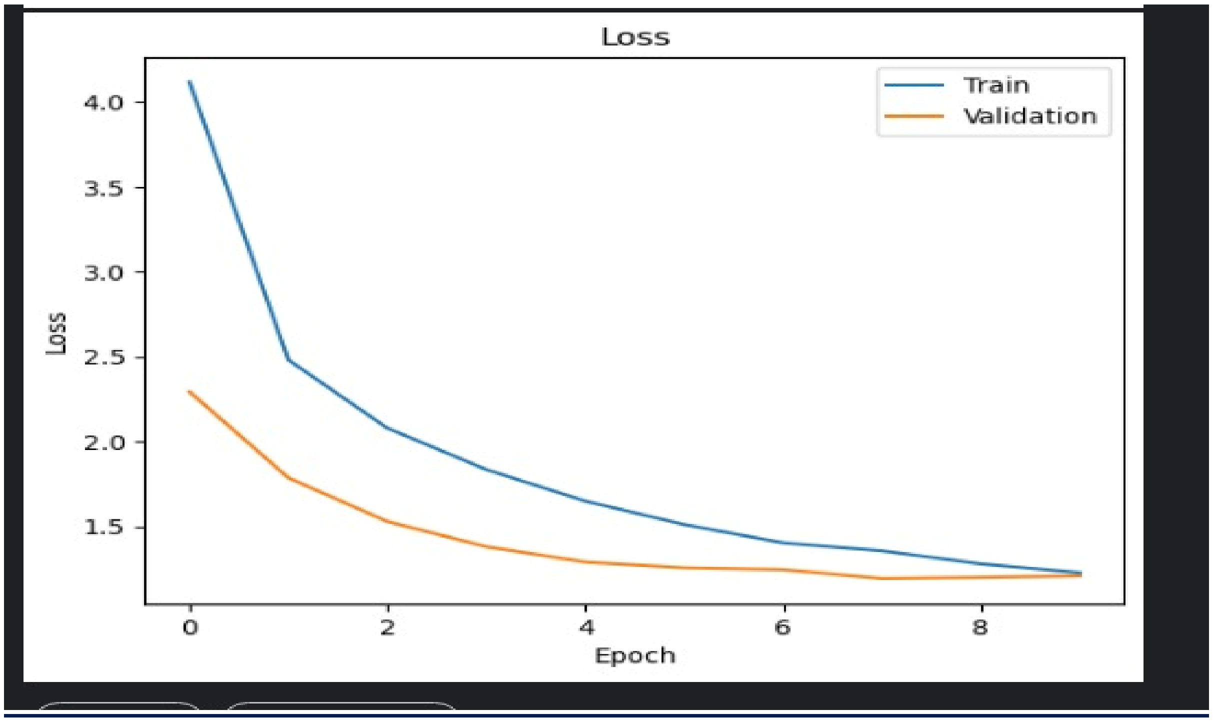
Output of VGG16



```
Epoch 1/10
359/359 [==============================] - 28s 74ms/step - loss: 4.1175 - accuracy: 0.2347 - val_loss: 2.2929 - val_accuracy:
0.4855
Epoch 2/10
359/359 [==============================] - 26s 70ms/step - loss: 2.4804 - accuracy: 0.4199 - val_loss: 1.7857 - val_accuracy:
0.5663
Epoch 3/10
359/359 [==============================] - 26s 72ms/step - loss: 2.0784 - accuracy: 0.4868 - val_loss: 1.5281 - val_accuracy:
0.6149
Epoch 4/10
359/359 [==============================] - 29s 79ms/step - loss: 1.8347 - accuracy: 0.5315 - val_loss: 1.3801 - val_accuracy:
0.6455
Epoch 5/10
359/359 [==============================] - 27s 74ms/step - loss: 1.6487 - accuracy: 0.5707 - val_loss: 1.2900 - val_accuracy:
0.6620
Epoch 6/10
359/359 [==============================] - 27s 73ms/step - loss: 1.5102 - accuracy: 0.5999 - val_loss: 1.2556 - val_accuracy:
0.6745
Epoch 7/10
359/359 [==============================] - 26s 72ms/step - loss: 1.4027 - accuracy: 0.6182 - val_loss: 1.2444 - val_accuracy:
0.6737
Epoch 8/10
359/359 [==============================] - 26s 70ms/step - loss: 1.3555 - accuracy: 0.6350 - val_loss: 1.1922 - val_accuracy:
0.6965
Epoch 9/10
359/359 [==============================] - 26s 72ms/step - loss: 1.2789 - accuracy: 0.6523 - val_loss: 1.1990 - val_accuracy:
0.6933
Epoch 10/10
359/359 [==============================] - 25s 69ms/step - loss: 1.2262 - accuracy: 0.6617 - val_loss: 1.2084 - val_accuracy:
0.6925
```

Plots

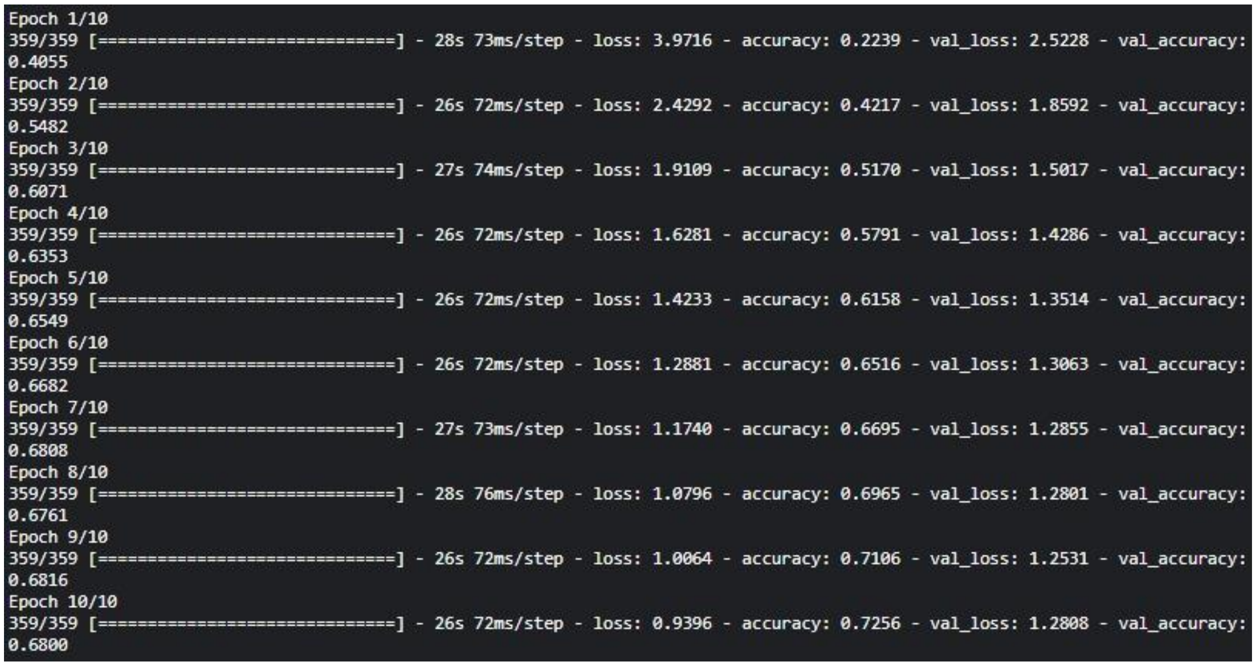

Accuracy
Train
Validation
0.7
0.6
0.5
0.4
0.3
Accuracy
0
2
4
6
8
Epoch


Loss
Train
Validation
4.0
3.5
3.0
2.5
2.0
1.5
Loss
0
2
4
6
8
Epoch

This model when we added dropout as 0.6 and number of filters as 512 with a relu.

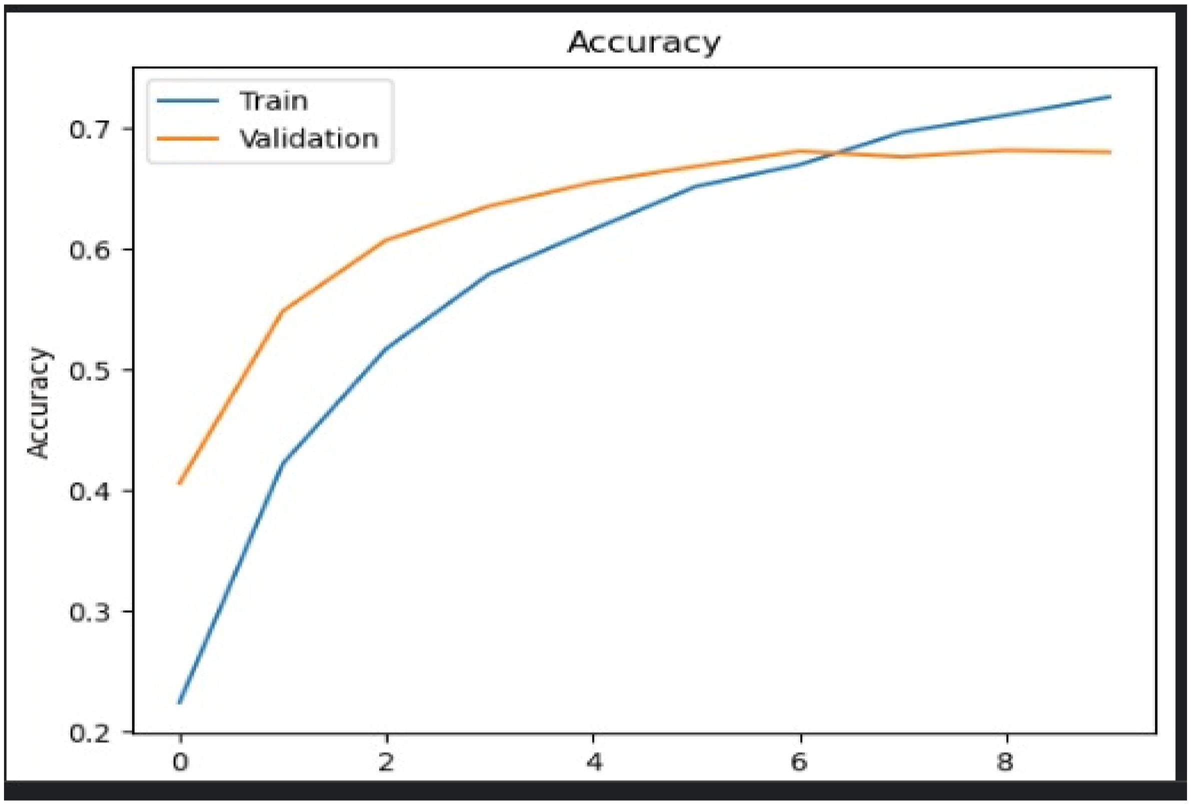
So as you can see the loss is decreasing but the accuracy is not high enough, but sure enough if we added more iterations the accuracy eventually will increase and get much accurate.

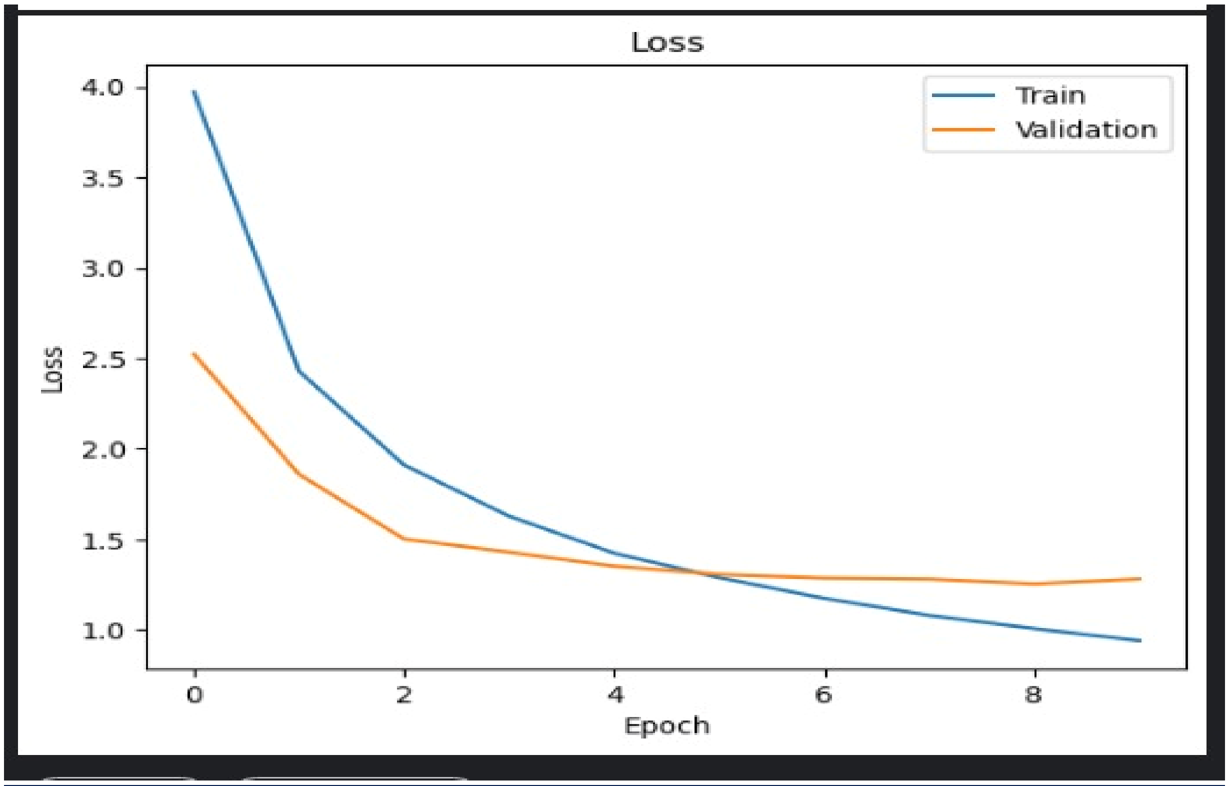
In the plots as you can see the accuracy the train and the validation are both increasing while the loss both are decreasing and here we tried to avoid the model to have overfitting.



```
Epoch 1/10
359/359 [==============================] - 28s 73ms/step - loss: 3.9716 - accuracy: 0.2239 - val_loss: 2.5228 - val_accuracy:
0.4055
Epoch 2/10
359/359 [==============================] - 26s 72ms/step - loss: 2.4292 - accuracy: 0.4217 - val_loss: 1.8592 - val_accuracy:
0.5482
Epoch 3/10
359/359 [==============================] - 27s 74ms/step - loss: 1.9109 - accuracy: 0.5170 - val_loss: 1.5017 - val_accuracy:
0.6071
Epoch 4/10
359/359 [==============================] - 26s 72ms/step - loss: 1.6281 - accuracy: 0.5791 - val_loss: 1.4286 - val_accuracy:
0.6353
Epoch 5/10
359/359 [==============================] - 26s 72ms/step - loss: 1.4233 - accuracy: 0.6158 - val_loss: 1.3514 - val_accuracy:
0.6549
Epoch 6/10
359/359 [==============================] - 26s 72ms/step - loss: 1.2881 - accuracy: 0.6516 - val_loss: 1.3063 - val_accuracy:
0.6682
Epoch 7/10
359/359 [==============================] - 27s 73ms/step - loss: 1.1740 - accuracy: 0.6695 - val_loss: 1.2855 - val_accuracy:
0.6808
Epoch 8/10
359/359 [==============================] - 28s 76ms/step - loss: 1.0796 - accuracy: 0.6965 - val_loss: 1.2801 - val_accuracy:
0.6761
Epoch 9/10
359/359 [==============================] - 26s 72ms/step - loss: 1.0064 - accuracy: 0.7106 - val_loss: 1.2531 - val_accuracy:
0.6816
Epoch 10/10
359/359 [==============================] - 26s 72ms/step - loss: 0.9396 - accuracy: 0.7256 - val_loss: 1.2808 - val_accuracy:
0.6800
```

<u>Plots</u>


Accuracy
Train
Validation
Accuracy
0.2
0.3
0.4
0.5
0.6
0.7
0
2
4
6
8


Loss
Train
Validation
Loss
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
2
4
6
8
Epoch

This is Vgg16 model but after we decreased the number of filters as 200 and also the dropout as 0.4, we took notice that the accuracy in epoch 10 here is 0.7256, while previously, when we did the number of filters as 512 and the drop out as 0.6 the accuracy in epoch 10 was 0.6617.

Also in the plots you can notice that there is a big difference from these 2 plots than the previous ones since we decrease here the dropout as 0.4 instead of 0.6 therefore it caused more overfitting.