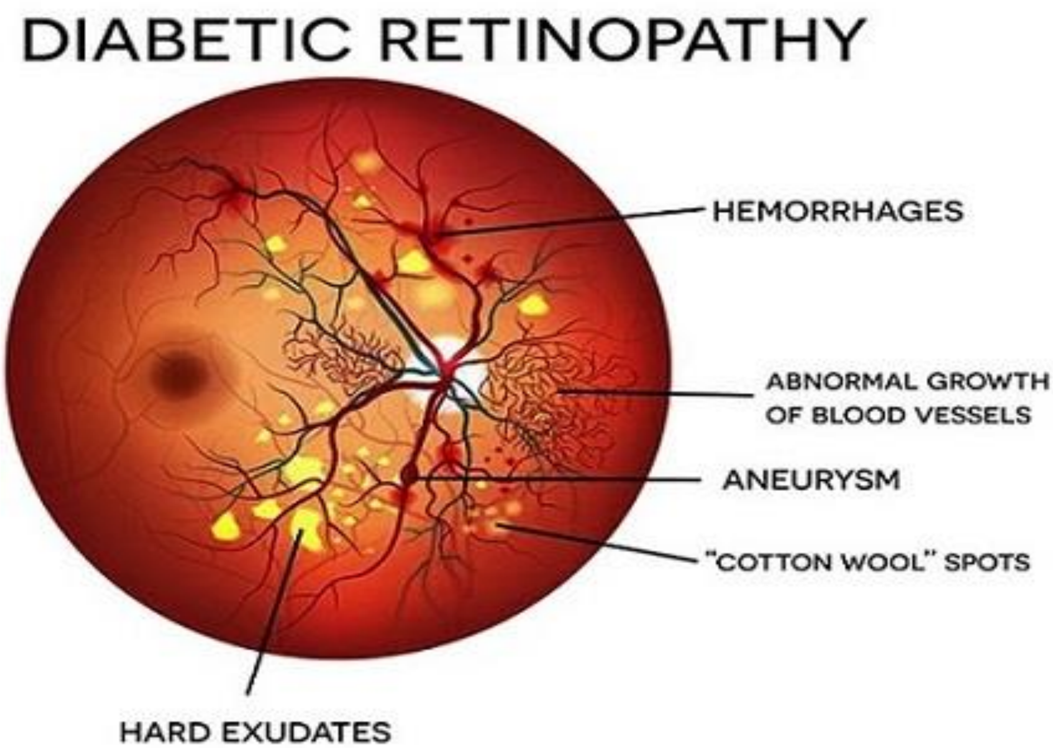
DIABETIC RETINOPATHY DETECTION

DEEP LEARNING PROJECT



S.KRISHNAN

subramoniankrishnan@gmail.com

INTRODUCTION

- Diabetic Retinopathy (DR) is a condition which develops in the eye, which if not treated at initial stages, could lead to **permanent blindness**.
- The World Health Organization estimates that 347 million people have the disease worldwide.
- India is said to be the diabetic capital of the world by 2030 with over 80 million people affected by it, and the prevalence of DR among persons with diabetes was 16.9% in India.
- There is a need for early identification of persons with diabetes retinopathy (DR) so that DR can be eliminated.
- For early and accurate detection of DR and stages of DR developing a DL model is required.

PROBLEM STATEMENT

Building a Deep Learning Model which can classify the given Images with a better Accuracy than the recorded Accuracy of **0.979** by the State of Art **Inception-Resnet-v2-FTCDW** Model.

META DATA

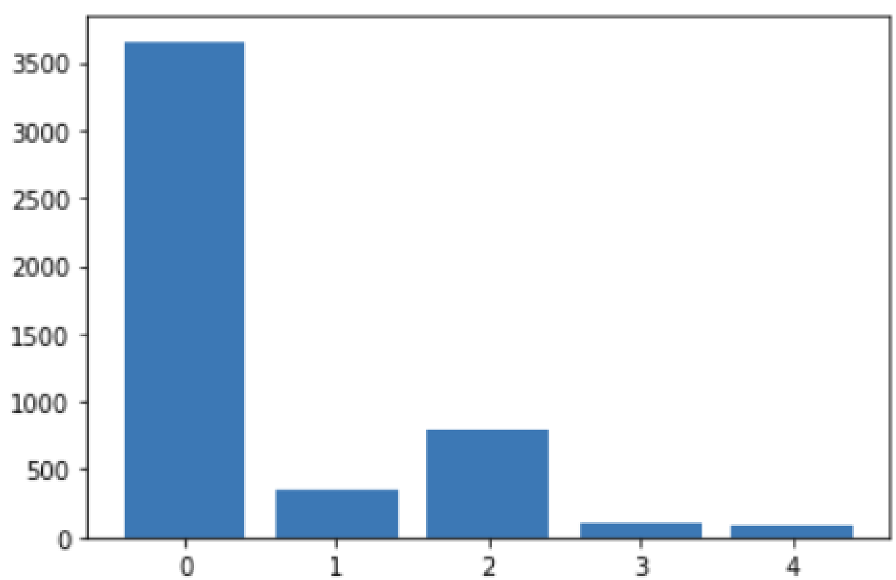
- The Dataset is called as EyePACS Dataset and the Retinal images were provided by EyePACS, a free platform for retinopathy screening.
- A large set of high-resolution retina images taken under a variety of imaging conditions. Left and right field is provided for every subject. Images are labeled with a subject id as well as either left or right (e.g., 1_left.jpeg is the left eye of patient id 1).
- It contains 88,702 high-resolution images with various resolutions, ranging from 433 × 289 pixels to 5184 × 3456 pixels, collected from different cameras.
- A clinician has rated the presence of diabetic retinopathy in each image on a scale of 0 to 4, according to the following scale:
- 0 - No DR
- 1 – Mild
- 2 – Moderate
- 3 - Severe
- 4 - Proliferative DR

The images in the dataset come from different models and types of cameras, which can affect the visual appearance of left vs. right. Some images are shown as one would see the retina anatomically (macula on the left, optic nerve on the right for the right eye). Others are shown as one would see through a microscope condensing lens (i.e., inverted, as one sees in a typical live eye exam). There are generally two ways to tell if an image is inverted:

- It is inverted if the macula (the small dark central area) is slightly higher than the midline through the optic nerve. If the macula is lower than the midline of the optic nerve, it's not inverted.
- If there is a notch on the side of the image (square, triangle, or circle) then it's not inverted. If there is no notch, it's inverted.

PRE-PROCESSING PIPELINE

- Diabetic Retinopathy Detection Dataset is a large dataset (around 88.29 GB) containing 5 zip files for training and 7 zip files for testing, we plan to use minibatches for training and testing as a technique to handle this large data.
- The Images are of High Resolution with their dimension's being around 4928 x 3256 (On Average) hence, we plan to Re-size the Image to a smaller dimension to speed-up the training process.
- After Exploring the data, we can see (Shown in the Image below) that there is a class Imbalance problem hence we will try to balance the distribution.



- We will use the Data Augmentation like Translating, Flipping and Rotating the Images to Increase the variety of the Images to be fed into the Network.
- Data Normalization technique used to Normalize the Images and convert them into a similar distribution.

OBJECTIVES

- We aim to Implement previous existing architectures ResNet, GoogleNet, AlexNet and get the accuracy about 95-97%, which is the recorded accuracies of the Models on EyePACS Dataset.
- As we know that a customized architecture performs better than existing architectures, we also try to implement it and improve the accuracy.

LITERATURE REVIEW

PAPER	YEAR	ACCURACY	ADVANTAGES
Explainable end-to-end deep learning for diabetic retinopathy detection across multiple datasets	2020	0.979	The proposed approach can robustly classify fundus images and detect DRs. The obtained results on these datasets show the higher performance obtained by our network and its generalization ability across multiple datasets. An explainability model was developed and showed that our model was able to efficiently identify different signs of DR

			and detect this health issue.
Deep Transfer Learning Models for Medical Diabetic Retinopathy Detection	2019	0.979	GoogleNet , VGG16, and VGG19. The mentioned CNN models have only a few layers when compared to large CNN models, such as Xception, DenseNet, and InceptionResNet, which consist of 71, 201 and 164 layers, respectively. The choice of these models reduces the training time and the complexity of the calculations.
Deep Learning based Early Detection and Grading of Diabetic Retinopathy Using Retinal Fundus Images	2018	0.98 (sensitivity) & 0.94 (specificity)	Without heavy data augmentation, a high-capacity network can easily overfit the training data. Even with dataaugmentation, any network can overfit on oversampled classes such as healthy eye (grade 0). Thus, designing a small capacity network with L2 regularization, and dropout has significant importance in retinopathy detection. So, in this work, we have presented a 4 x 4 kernel-based CNN network with several preprocessing and augmentation methods to improve

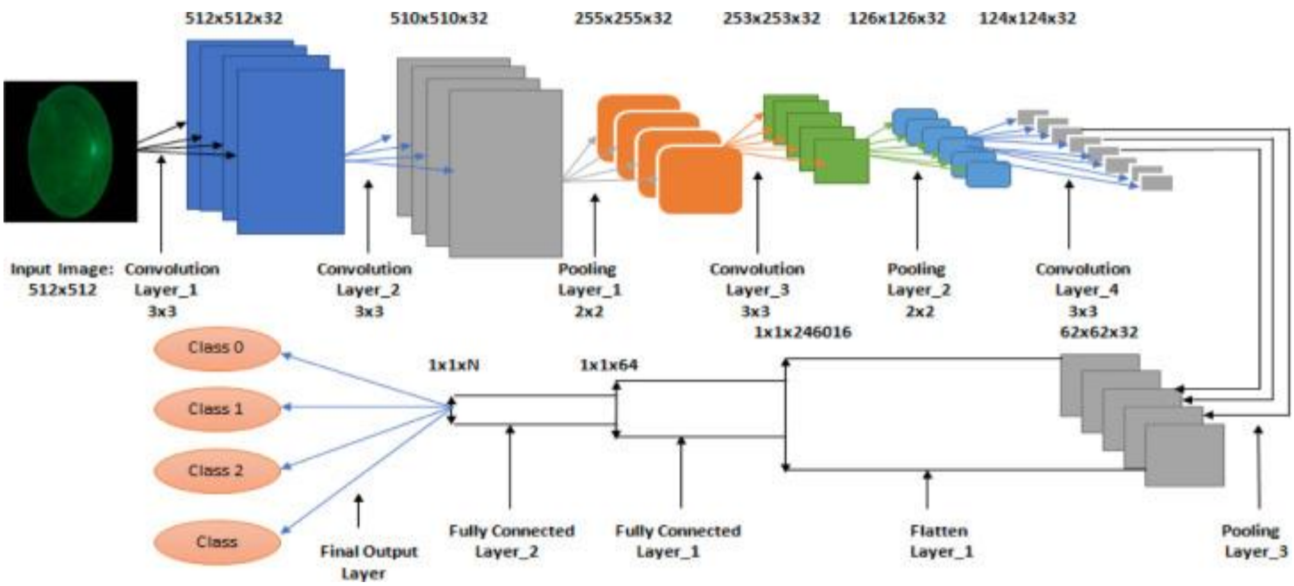
			the performance of the architecture.
Transfer Learning based Detection of Diabetic Retinopathy from Small Dataset	2019	0.909	Transfer learning is implemented to classify DR into 2 classes with a much-reduced training data than other previous DR classification techniques employed. This was done to design a way to train a DL model that performs well on unseen data by efficiently learning from small dataset because training data is limited in healthcare.
Deep learning frameworks for diabetic retinopathy detection with smartphone-based retinal imaging systems	2020	0.986	This paper presented the utility of CNN-based AlexNet, GoogLeNet, and ResNet50 frameworks to improve the performance of DR detection in smartphone-based and traditional fundus camera retina images.

MODELS TO BE IMPLEMENTED

AlexNet, InceptionV3, InceptionResNetV2

BASELINE MODEL

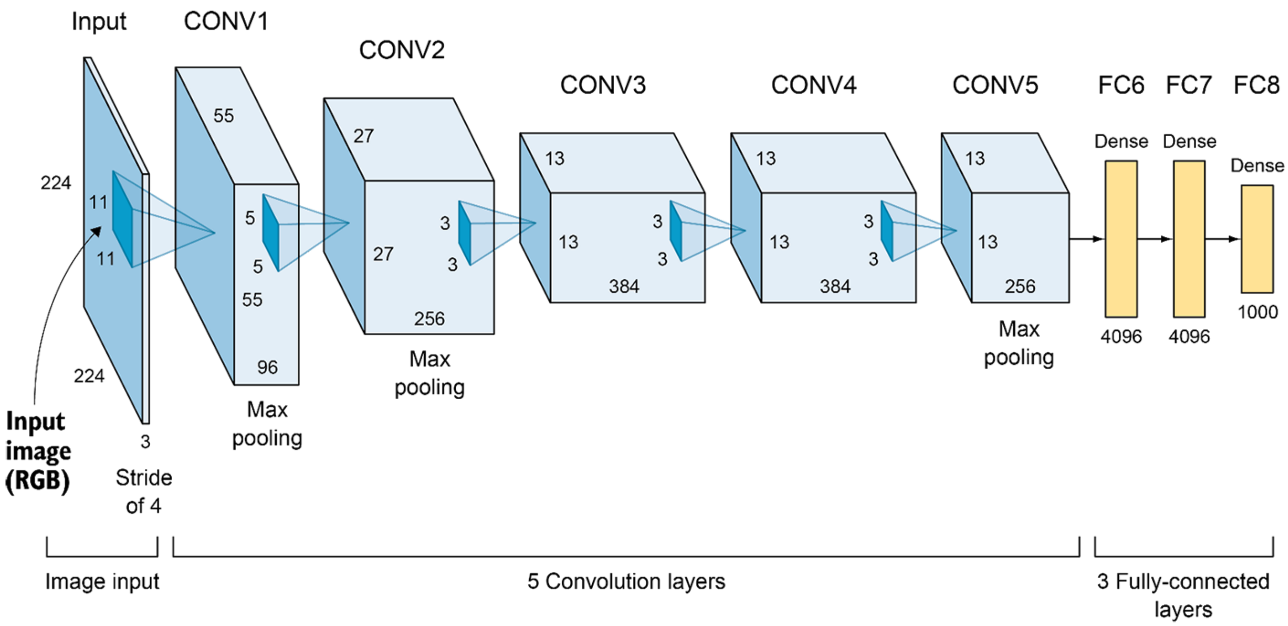
AlexNet: -



Structural Details: -

Layer Name	Tensor Size	Weights	Biases	Parameters
Input Image	227x227x3	0	0	0
Conv-1	55x55x96	34,848	96	34,944
MaxPool-1	27x27x96	0	0	0
Conv-2	27x27x256	614,400	256	614,656
MaxPool-2	13x13x256	0	0	0
Conv-3	13x13x384	884,736	384	885,120
Conv-4	13x13x384	1,327,104	384	1,327,488
Conv-5	13x13x256	884,736	256	884,992
MaxPool-3	6x6x256	0	0	0
FC-1	4096×1	37,748,736	4,096	37,752,832
FC-2	4096×1	16,777,216	4,096	16,781,312
FC-3	1000×1	4,096,000	1,000	4,097,000
Output	1000×1	0	0	0
Total				**62,378,344**

Architecture: -



RESULTS

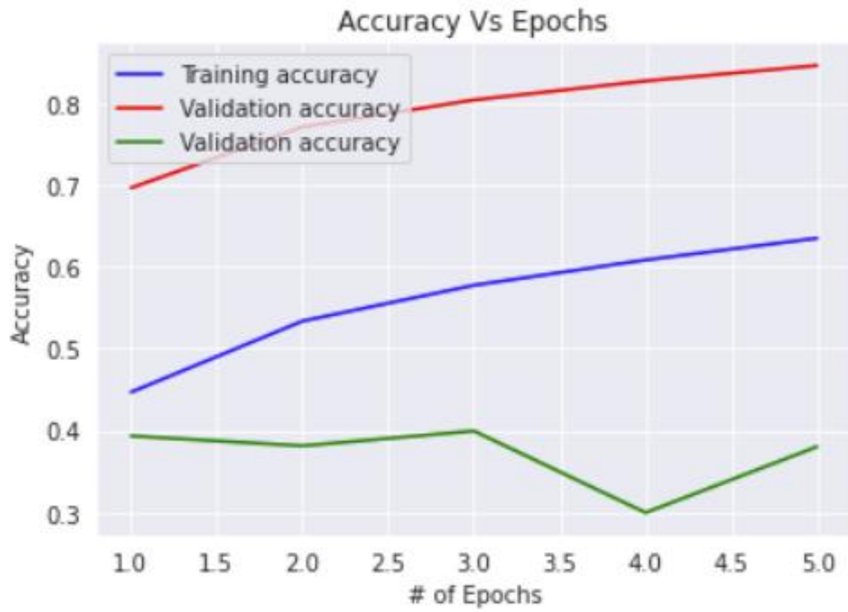
Metrics \ Models	InceptionV3	InceptionResNetV2	AlexNet
Categorical Accuracy	36.11	30.37	51.73
Validation Categorical Accuracy	40.48	3.19	37.24
Top 2 Accuracy	77.53	54.86	73.80
Validation Top 2 Accuracy	82.34	68.69	69.68

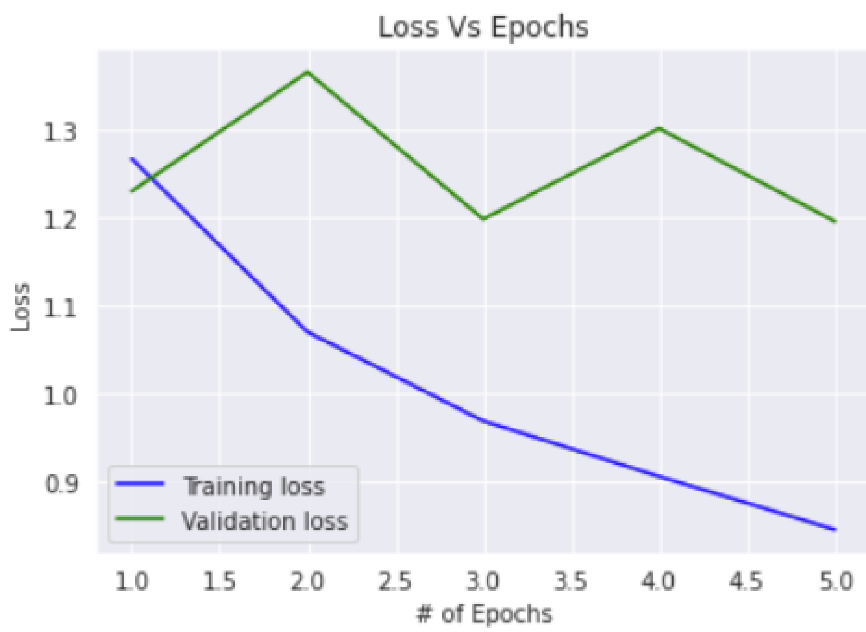
Loss \ Models	InceptionV3	InceptionResNetV2	AlexNet
Loss	1.18	1.56	1.77
Validation Loss	1.20	1.60	2.03

Hyperparameters \ Models	InceptionV3	InceptionResNetV2	AlexNet
Epochs	10	12	15
Learning Rate	0.001	0.001	0.0001
Batch Size	64	64	64
Optimizer	adam	adam	adam

PLOTS GENERATED

InceptionV3



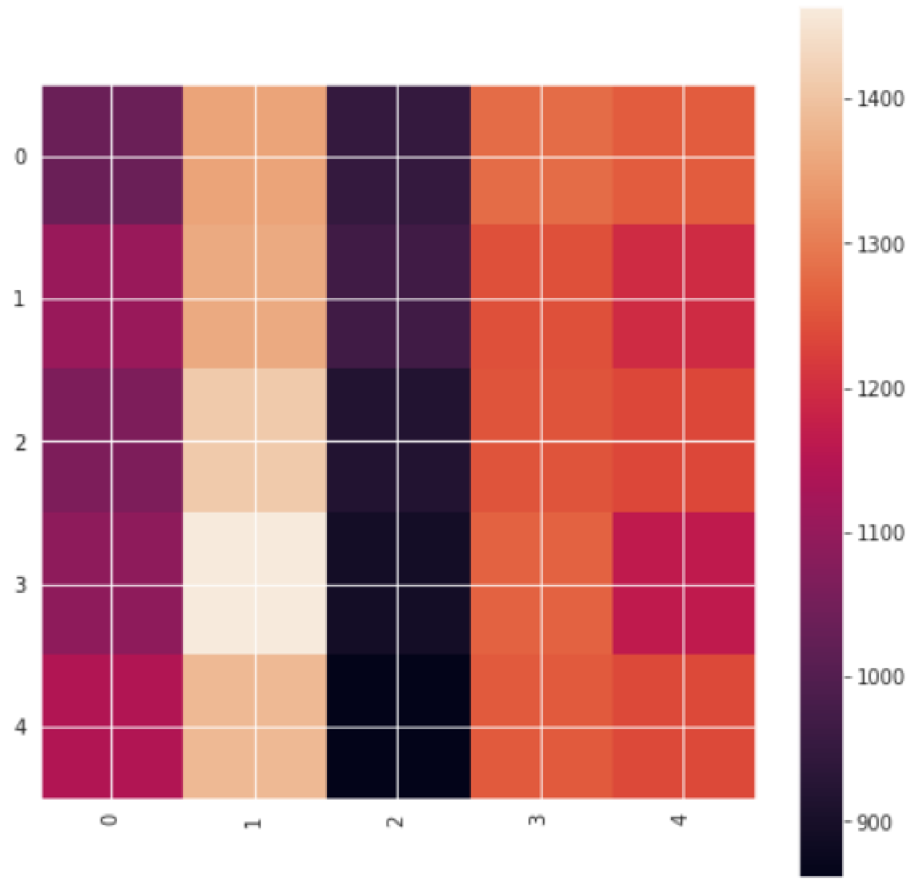


Classification Report:

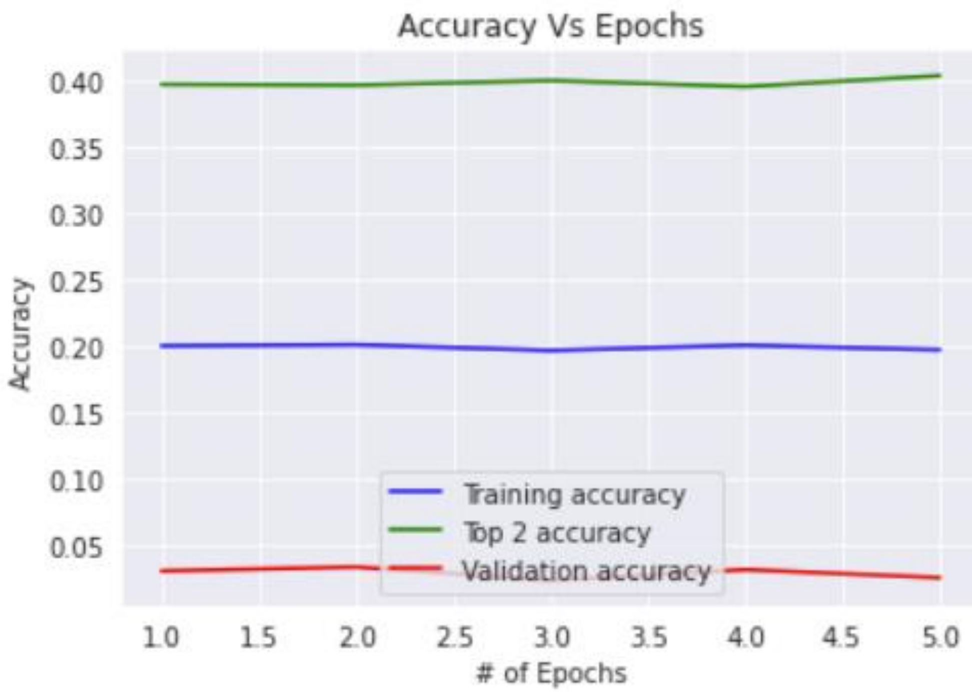
	precision	recall	f1-score	support
0	0.19	0.18	0.18	5883
1	0.20	0.23	0.21	5883
2	0.20	0.16	0.18	5883
3	0.20	0.22	0.21	5883
4	0.20	0.21	0.21	5883
accuracy			0.20	29415
macro avg	0.20	0.20	0.20	29415
weighted avg	0.20	0.20	0.20	29415

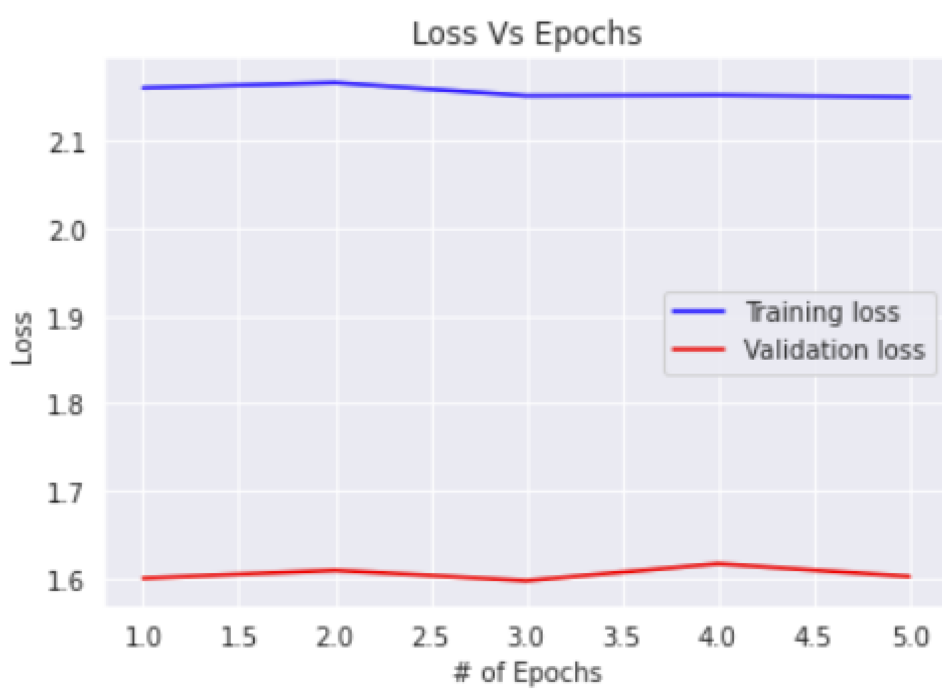
Confusion Matrix:

```
[[1037 1355  951 1279 1261]
 [1107 1363  969 1246 1198]
 [1066 1412  920 1251 1234]
 [1092 1464  893 1268 1166]
 [1142 1385  862 1258 1236]]
```


0
1
2
3
4
0
1
2
3
4
1400
1300
1200
1100
1000
900

InceptionResNetV2



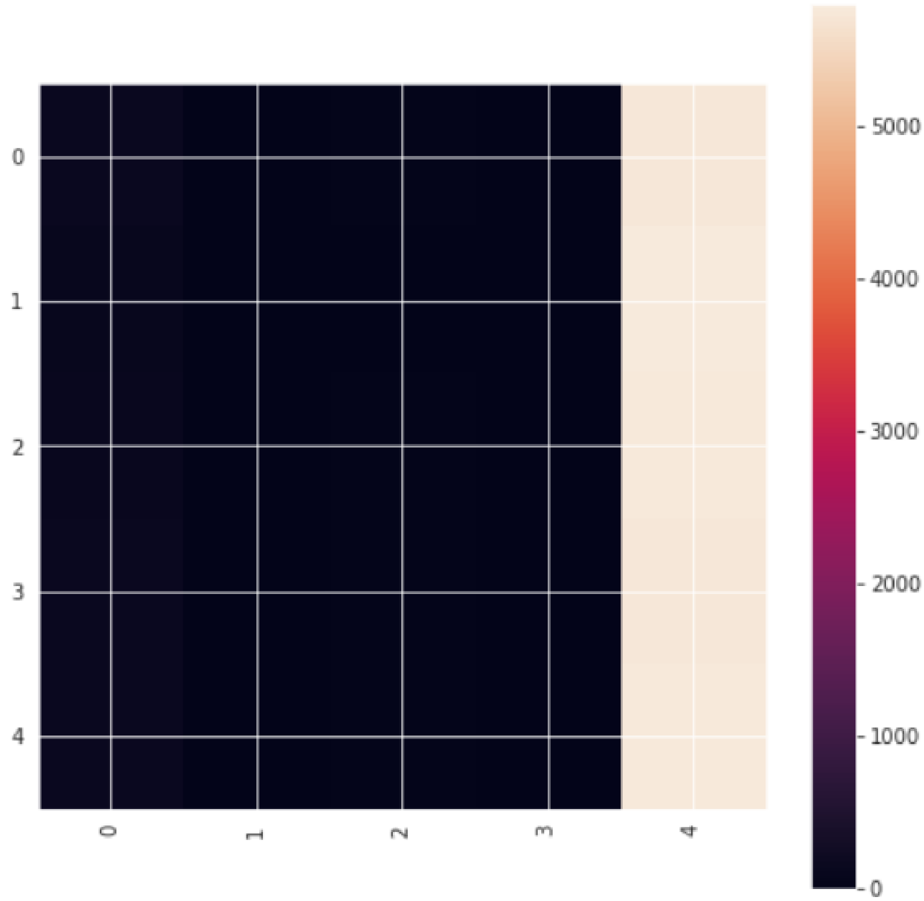


Classification Report

	precision	recall	f1-score	support
0	0.21	0.02	0.04	5937
1	0.38	0.00	0.00	5937
2	0.20	0.01	0.01	5937
3	0.00	0.00	0.00	5937
4	0.20	0.97	0.33	5937
accuracy			0.20	29685
macro avg	0.20	0.20	0.08	29685
weighted avg	0.20	0.20	0.08	29685

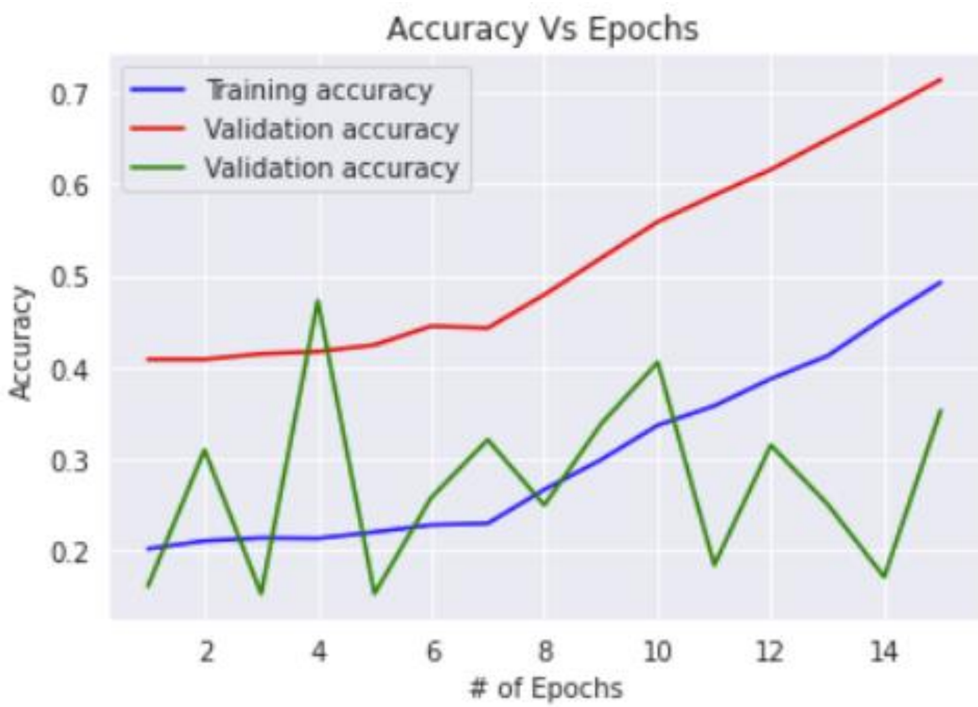
Confusion Matrix

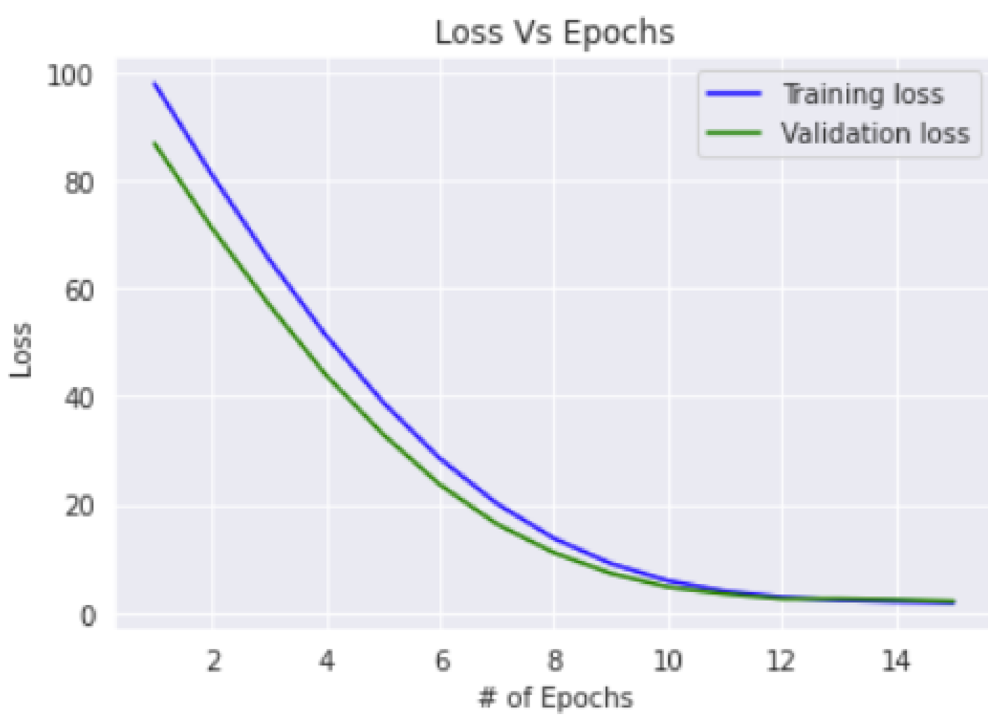
```
[[ 141    2   43    0 5751]
 [ 112    3   21    0 5801]
 [ 135    3   34    0 5765]
 [ 143    0   41    0 5753]
 [ 140    0   35    0 5762]]
```



0
1
2
3
4
0
1
2
3
4
5000
4000
3000
2000
1000
0

AlexNet


Accuracy Vs Epochs
Training accuracy
Validation accuracy
Validation accuracy
0.7
0.6
0.5
0.4
0.3
0.2
Accuracy
2
4
6
8
10
12
14
# of Epochs

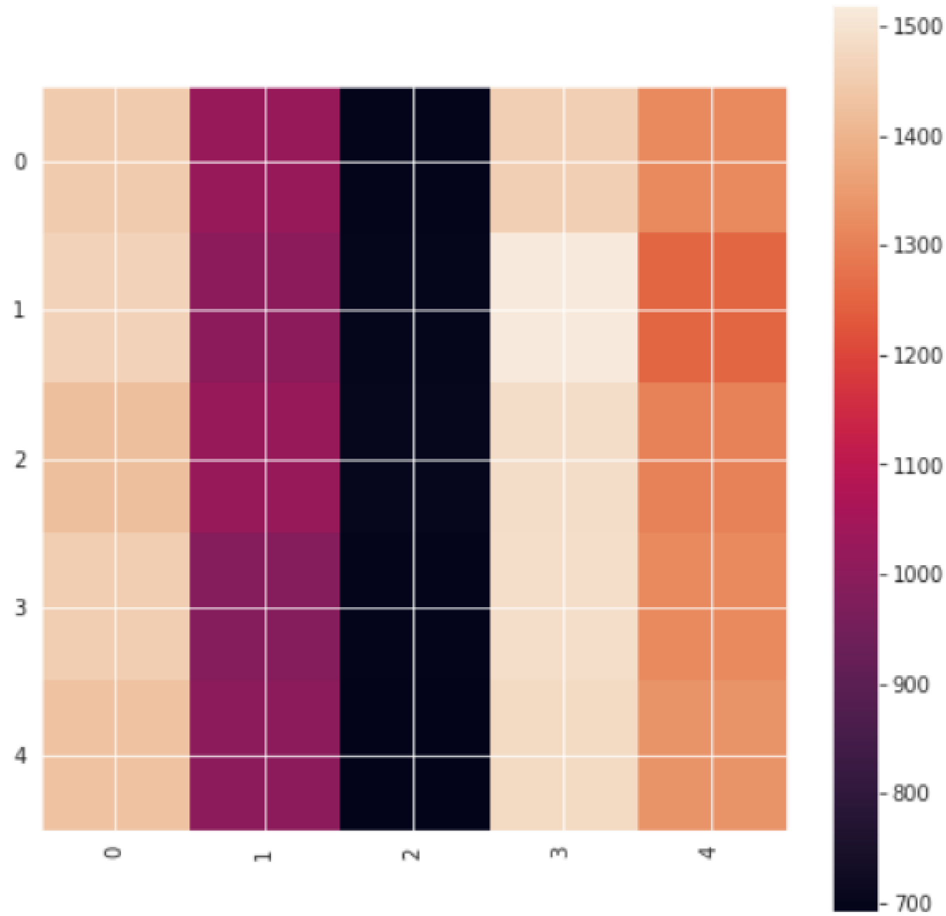

Loss Vs Epochs
Training loss
Validation loss
100
80
60
40
20
0
Loss
2
4
6
8
10
12
14
# of Epochs

Classification Report:

	precision	recall	f1-score	support
0	0.20	0.24	0.22	5946
1	0.20	0.17	0.18	5946
2	0.20	0.12	0.15	5946
3	0.20	0.25	0.22	5946
4	0.20	0.22	0.21	5946
accuracy			0.20	29730
macro avg	0.20	0.20	0.20	29730
weighted avg	0.20	0.20	0.20	29730

Confusion Matrix:

```
[[1448 1025  697 1459 1317]
 [1467 1004  701 1519 1255]
 [1423 1027  704 1487 1305]
 [1456  987  696 1491 1316]
 [1434 1001  692 1483 1336]]
```



0
1
2
3
4
0
1
2
3
4
1500
1400
1300
1200
1100
1000
900
800
700

CONCLUSION

By analyzing the above results, we concluded that InceptionV3 performed the best. The training and validation accuracy of the model is 86.8 and 81.3, Also it has the least loss when compared to the other models.

InceptionV3 has also learned to represent the data very well unlike other models. So, InceptionV3 along with the hyperparameters: Epochs = 10, Batch Size = 64, Optimizers = Adam, learning rate = 0.001 performs the best on the given data. It is evaluated with the metrics: Categorical Accuracy and Top 2 Accuracy.

Furthermore, we will try to implement the Visual Attention mechanism with some tweaking to the baseline InceptionV3 architecture so that it can perform better. We also plan to implement data augmentation and increase the epochs so that our model can learn and generalize better.