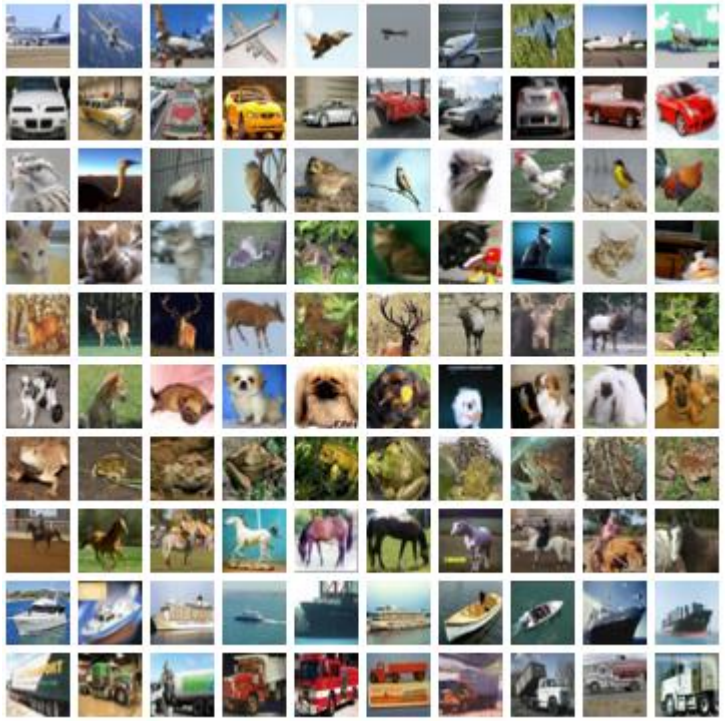
CIFAR
CANADIAN INSTITUTE FOR ADVANCED RESEARCH

FINAL CAPSTONE PROJECT

CIFAR10 Dataset

Image Recognition using Convolutional Neural networks



Presented by: Usman Shaikh

PROJECT DESCRIPTION AND PURPOSE

Description

- Dataset: https://www.cs.toronto.edu/~kriz/cifar.html
- CIFAR-10 is a subset of the 80 million tiny images dataset created by **Canadian Institute for Advanced Research (CIFAR)**.
- The CIFAR-10 dataset consists of 60000 32x32x3 color images in 10 equal classes, (6000 images per class).
- Each class of images corresponds to a physical object (automobile, cat, dog, airplane, etc.)
- In total there are 10 classes including airplane, automobile, bird, cat, deer, dog, frog, horse, ship and a truck.

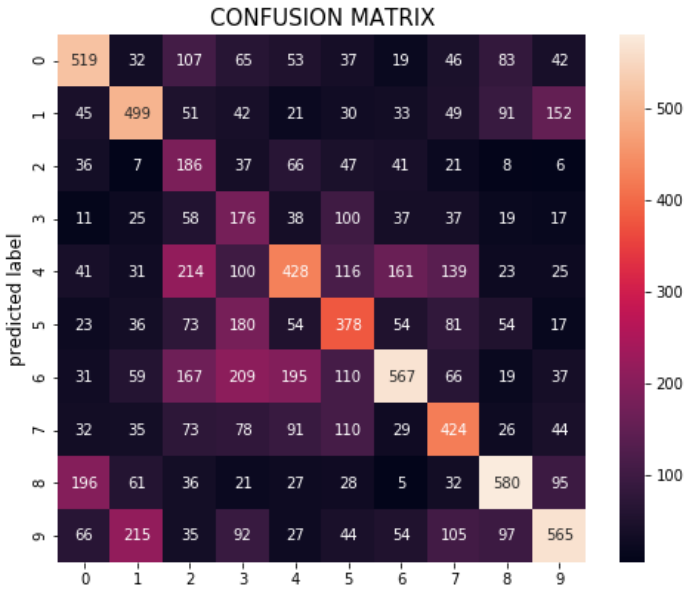
Problem Summary

- The purpose of this project is to develop a multi class classification model successfully predicting different classes.
- Our goal is to train and test different machine learning models as per our requirement and then come up with one specific model which provides us with the best possible and consistent results in a timely fashion.
- For this purpose we will be building eleven different machine learning models, two each for the Random Forest, Logistic Regression and Gradient Boost followed by 6 separate models of Convolutional Neural Networks.

RANDOM FOREST

Without PCA

- Variables: 3,072
- Estimators: 500
- Max depth: 10
- R^2 Train Data: 0.64
- R^2 Test Data: 0.43
- Mean Absolute Error: 2.03
- Mean Squared Error: 10.67
- Root Mean Squared Error: 3.26
- Time: 00:08:16.12

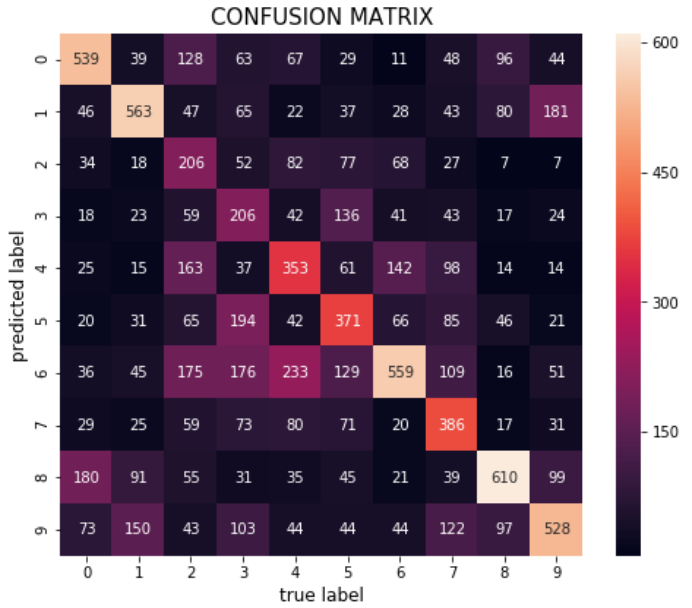


Classification Report Testing Set

	precision	recall	f1-score	support
0	0.52	0.52	0.52	1000
1	0.49	0.50	0.50	1000
2	0.41	0.19	0.26	1000
3	0.34	0.18	0.23	1000
4	0.33	0.43	0.38	1000
5	0.40	0.38	0.39	1000
6	0.39	0.57	0.46	1000
7	0.45	0.42	0.44	1000
8	0.54	0.58	0.56	1000
9	0.43	0.56	0.49	1000
micro avg	0.43	0.43	0.43	10000
macro avg	0.43	0.43	0.42	10000
weighted avg	0.43	0.43	0.42	10000

With PCA

- Components = 56
- Estimators: 500
- Max depth: 10
- R^2 Train Data: 0.60
- R^2 Test Data: 0.43
- Mean Absolute Error: 2.05
- Mean Squared Error: 10.66
- Root Mean Squared Error: 3.26
- Time: 00:01:53.35



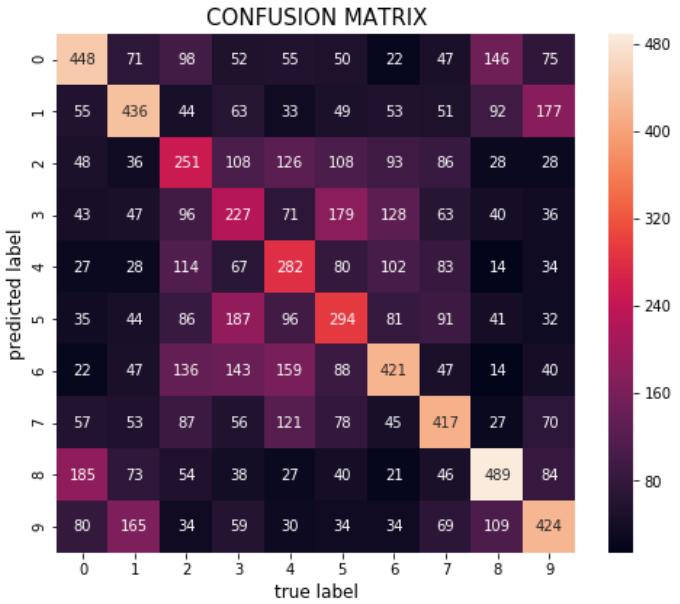
Classification Report Testing Set

	precision	recall	f1-score	support
0	0.51	0.54	0.52	1000
1	0.51	0.56	0.53	1000
2	0.36	0.21	0.26	1000
3	0.34	0.21	0.26	1000
4	0.38	0.35	0.37	1000
5	0.39	0.37	0.38	1000
6	0.37	0.56	0.44	1000
7	0.49	0.39	0.43	1000
8	0.51	0.61	0.55	1000
9	0.42	0.53	0.47	1000
micro avg	0.43	0.43	0.43	10000
macro avg	0.43	0.43	0.42	10000
weighted avg	0.43	0.43	0.42	10000

LOGISTIC REGRESSION

Without PCA

- Variables: 3,072
- R^2 Train Data: 0.52
- R^2 Test Data: 0.36
- Mean Absolute Error: 2.31
- Mean Squared Error: 12.20
- Root Mean Squared Error: 3.49
- Time: 01:49:33.14

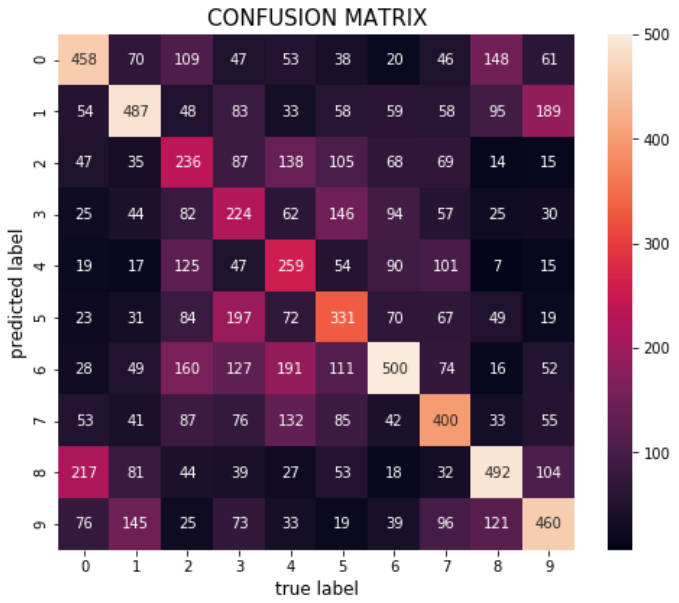


Classification Report Testing Set

	precision	recall	f1-score	support
0	0.42	0.45	0.43	1000
1	0.41	0.44	0.42	1000
2	0.28	0.25	0.26	1000
3	0.24	0.23	0.24	1000
4	0.34	0.28	0.31	1000
5	0.30	0.29	0.30	1000
6	0.38	0.42	0.40	1000
7	0.41	0.42	0.41	1000
8	0.46	0.49	0.48	1000
9	0.41	0.42	0.42	1000
micro avg	0.37	0.37	0.37	10000
macro avg	0.37	0.37	0.37	10000
weighted avg	0.37	0.37	0.37	10000

With PCA

- Components = 56
- R^2 Train Data: 0.3832
- R^2 Test Data: 0.38
- Mean Absolute Error: 2.25
- Mean Squared Error: 11.86
- Root Mean Squared Error: 3.44
- Time: 00:00:16.38



Classification Report Testing Set

	precision	recall	f1-score	support
0	0.44	0.46	0.45	1000
1	0.42	0.49	0.45	1000
2	0.29	0.24	0.26	1000
3	0.28	0.22	0.25	1000
4	0.35	0.26	0.30	1000
5	0.35	0.33	0.34	1000
6	0.38	0.50	0.43	1000
7	0.40	0.40	0.40	1000
8	0.44	0.49	0.47	1000
9	0.42	0.46	0.44	1000
micro avg	0.38	0.38	0.38	10000
macro avg	0.38	0.38	0.38	10000
weighted avg	0.38	0.38	0.38	10000

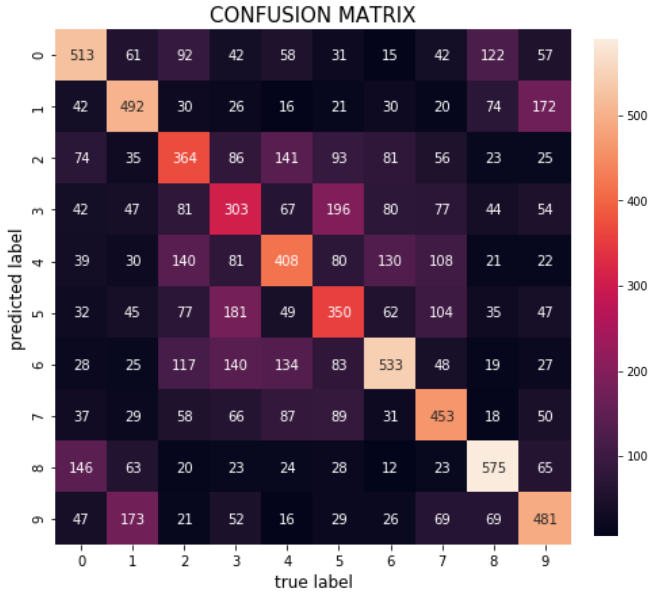
GRADIENT BOOST

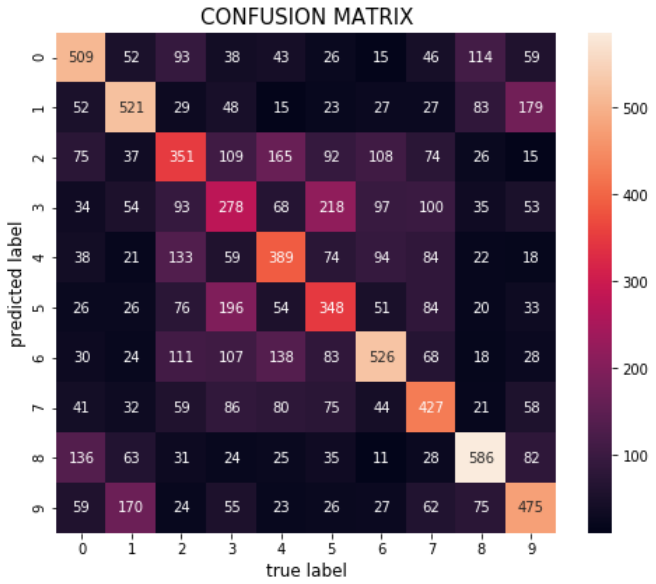
Without PCA

- Variables: 3,072
- R^2 Train Data: 0.94
- R^2 Test Data: 0.44
- Mean Absolute Error: 1.98
- Mean Squared Error: 10.21
- Root Mean Squared Error: 3.19
- Subsample = 0.3
- Estimators = 50
- Max depth =10
- Time: 09:24:40.93

With PCA

- Components = 56
- R^2 Train Data: 0.92
- R^2 Test Data: 0.44
- Mean Absolute Error: 2.01
- Mean Squared Error: 10.35
- Root Mean Squared Error: 3.21
- Subsample = 0.3
- Estimators = 50
- Max depth =10
- Time: 00:14:35.99





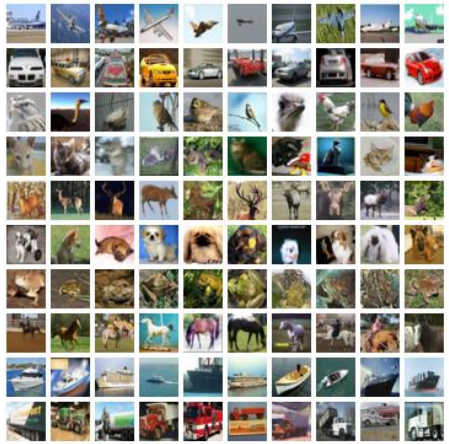
Classification Report Testing Set

	precision	recall	f1-score	support
0	0.51	0.51	0.51	1000
1	0.52	0.52	0.52	1000
2	0.33	0.35	0.34	1000
3	0.27	0.28	0.27	1000
4	0.42	0.39	0.40	1000
5	0.38	0.35	0.36	1000
6	0.46	0.53	0.49	1000
7	0.46	0.43	0.44	1000
8	0.57	0.59	0.58	1000
9	0.48	0.47	0.48	1000
micro avg	0.44	0.44	0.44	10000
macro avg	0.44	0.44	0.44	10000
weighted avg	0.44	0.44	0.44	10000

MODEL COMPARISON

	Random Forest	Logistic Regression	Gradient boost
Without PCA	R^2 Train Data: 0.64 R^2 Test Data: 0.43	R^2 Train Data: 0.52 R^2 Test Data: 0.36	R^2 Train Data: 0.94 R^2 Test Data: 0.44
Time	00:08:16.12	01:49:33.14	09:24:40.93
With PCA	R^2 Train Data: 0.60 R^2 Test Data: 0.43	R^2 Train Data: 0.38 R^2 Test Data: 0.38	R^2 Train Data: 0.92 R^2 Test Data: 0.44
Time	00:01:53.35	00:00:16.38	00:09:43.77

- For comparison model parameters are kept same for both instances.
- It shows that we are better off using Principle Component Analysis (PCA).
- Processing times can be considerably reduced without compromising accuracy.
- Models still require further tuning and fail to produce desired results.



CONVOLUTIONAL NEURAL NETWORKS – MODEL 1

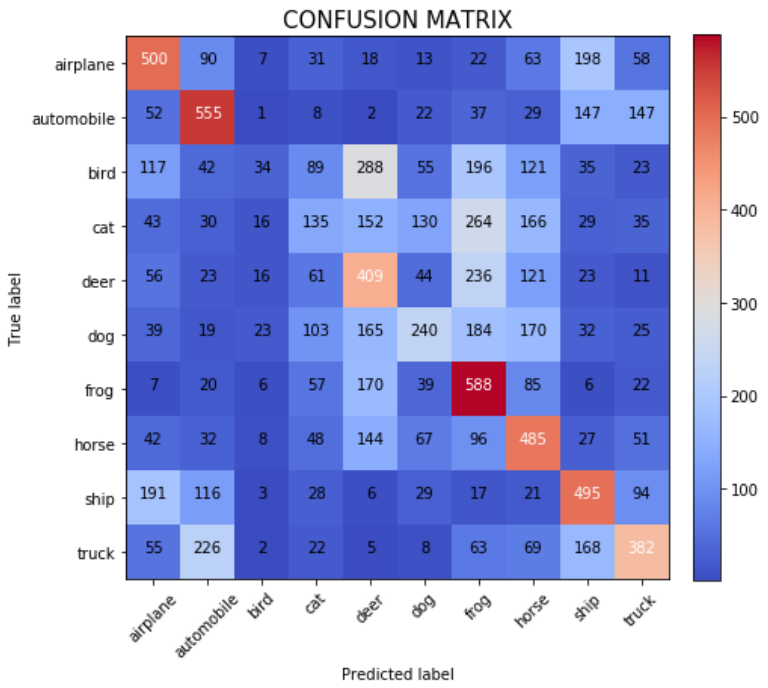
MODEL 1 (4 Layers)

Name: model1

- Variables: 3,072
- Convolutional layers: 4
- Max pooling layers: 2
- Dense Layers: 2
- Dropout layers: 3 (0.25 – 0.5)
- Activation Functions used
  - Rectified Linear Unit (RELU)
  - Soft max
- Optimizer: SGD
- Loss = sparse categorical cross entropy
- Learning rate = 0.0001
- Batch size = 32
- Epochs = 100
- Total time: 02:29:19.31

Test loss: 1.72

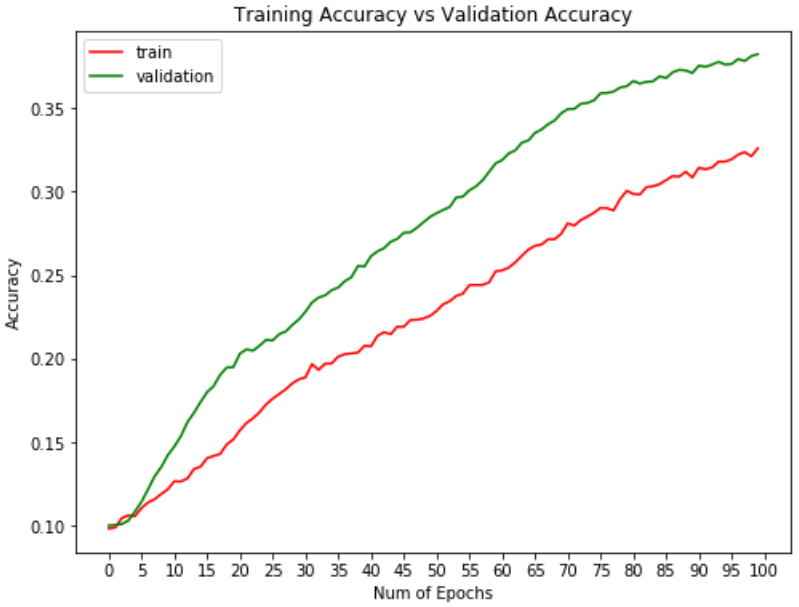
Test accuracy: 0.38

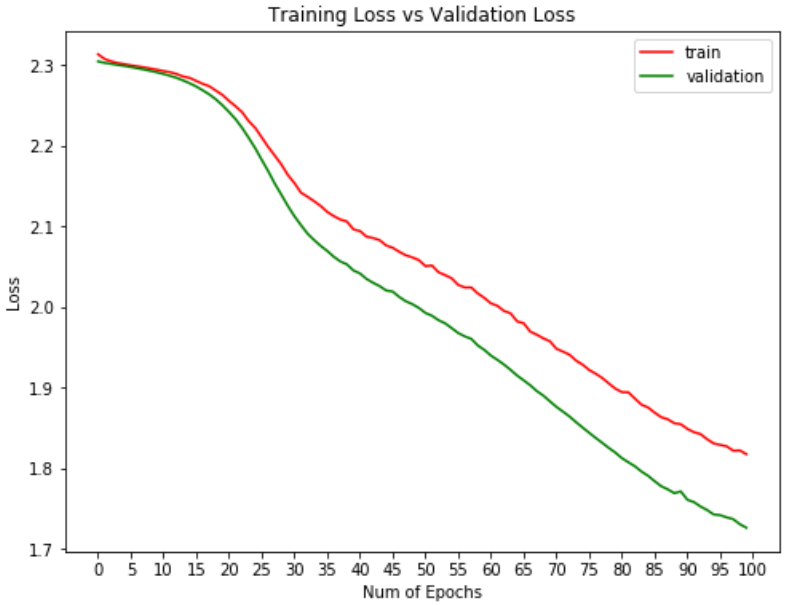



```
Classification Report
              precision    recall  f1-score   support

           0       0.45      0.50      0.48      1000
           1       0.48      0.56      0.52      1000
           2       0.29      0.03      0.06      1000
           3       0.23      0.14      0.17      1000
           4       0.30      0.41      0.35      1000
           5       0.37      0.24      0.29      1000
           6       0.35      0.59      0.44      1000
           7       0.36      0.48      0.42      1000
           8       0.43      0.49      0.46      1000
           9       0.45      0.38      0.41      1000

   micro avg       0.38      0.38      0.38     10000
   macro avg       0.37      0.38      0.36     10000
weighted avg       0.37      0.38      0.36     10000
```





CONVOLUTIONAL NEURAL NETWORKS – MODEL 2

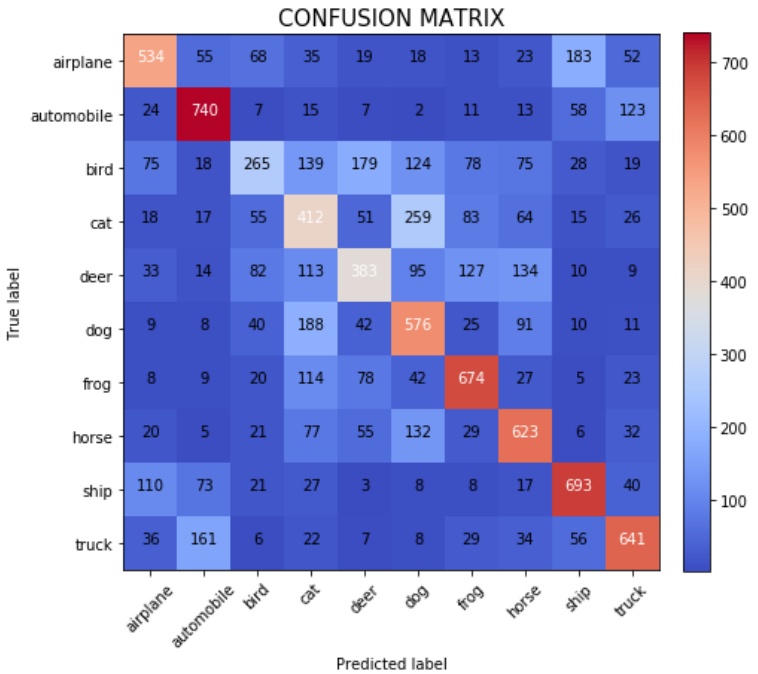
MODEL 2 (4 Layers)

Name: *model1b*

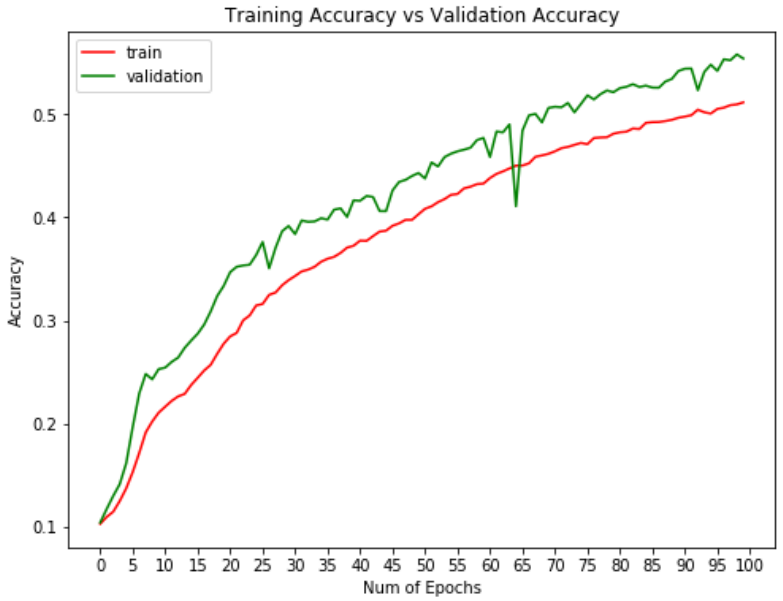
- Variables: 3,072
- Convolutional layers: 4
- Max pooling layers: 2
- Dense Layers: 2
- Dropout layers: 3 (0.25 – 0.5)
- Activation Functions used
  - Rectified Linear Unit (RELU)
  - Soft max
- Optimizer: SGD
- Loss = sparse categorical cross entropy
- Learning rate = 0.001 +
- Batch size = 64 +
- Epochs = 100
- Total time: 02:18:07.45

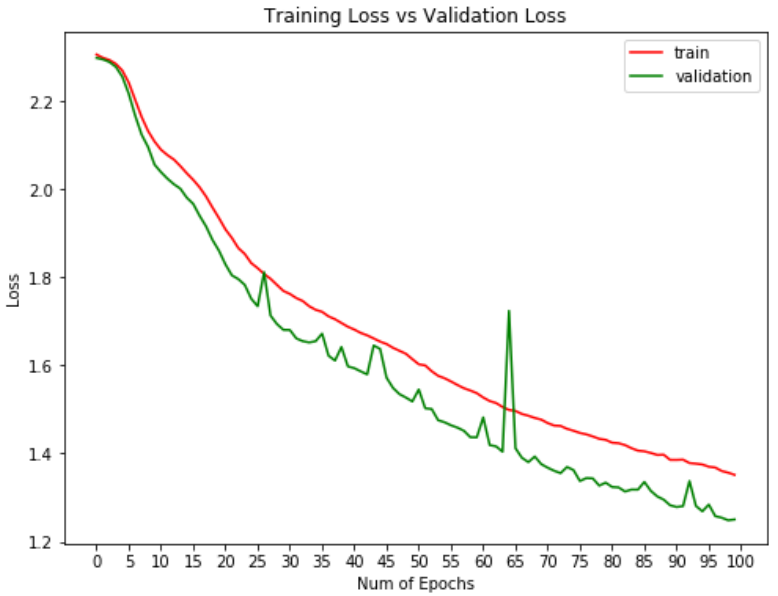
Test loss: 1.25

Test accuracy: 0.55



Classification Report	precision	recall	f1-score	support
0	0.62	0.53	0.57	1000
1	0.67	0.74	0.70	1000
2	0.45	0.27	0.33	1000
3	0.36	0.41	0.38	1000
4	0.46	0.38	0.42	1000
5	0.46	0.58	0.51	1000
6	0.63	0.67	0.65	1000
7	0.57	0.62	0.59	1000
8	0.65	0.69	0.67	1000
9	0.66	0.64	0.65	1000
micro avg	0.55	0.55	0.55	10000
macro avg	0.55	0.55	0.55	10000
weighted avg	0.55	0.55	0.55	10000





CONVOLUTIONAL NEURAL NETWORKS – MODEL 3

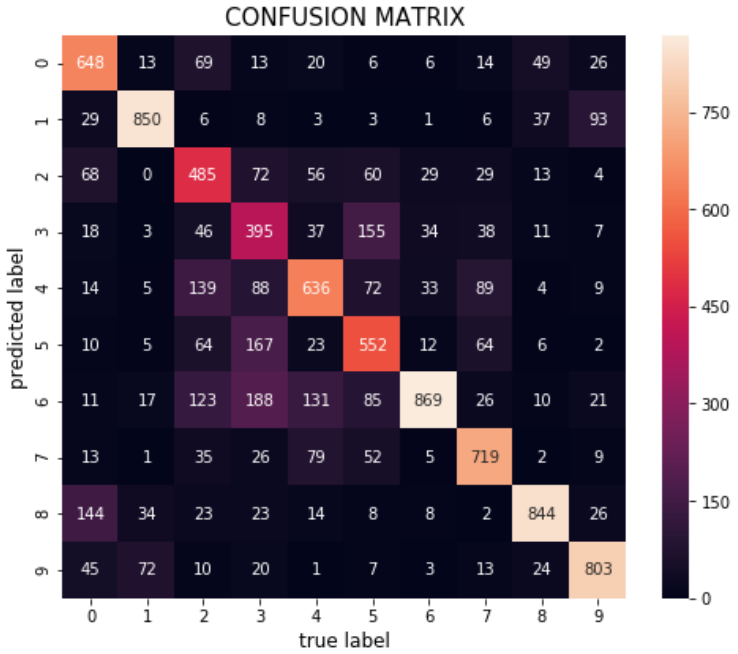
MODEL 3 (6 Layers)

Name: *model2b*

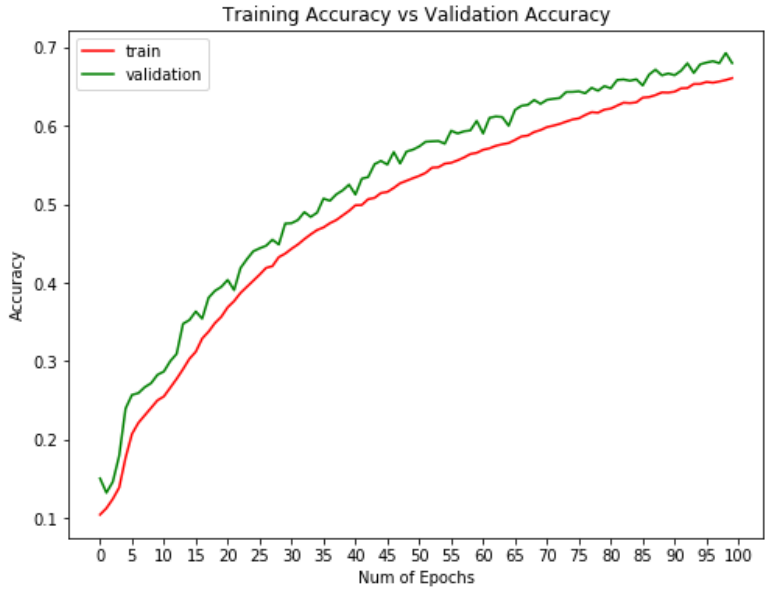
- Variables: 3,072
- Convolutional layers: 6 +
- Max pooling layers: 3 +
- Dense Layers: 2
- Dropout layers: 4 (0.25 – 0.5) +
- Activation Functions used
  - Rectified Linear Unit (RELU)
  - Soft max
- Optimizer: SGD
- Loss = sparse categorical cross entropy
- Learning rate = 0.001
- Batch size = 32
- Epochs = 100 +
- Total time: 04:27:32.18

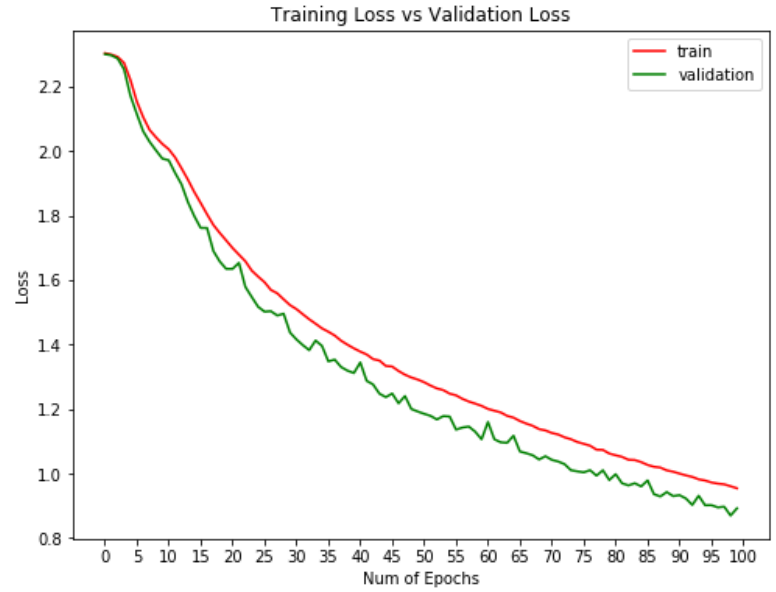
Test loss: 0.89

Test accuracy: 0.68



Classification Report	precision	recall	f1-score	support
0	0.75	0.65	0.70	1000
1	0.82	0.85	0.83	1000
2	0.59	0.48	0.53	1000
3	0.53	0.40	0.45	1000
4	0.58	0.64	0.61	1000
5	0.61	0.55	0.58	1000
6	0.59	0.87	0.70	1000
7	0.76	0.72	0.74	1000
8	0.75	0.84	0.79	1000
9	0.80	0.80	0.80	1000
micro avg	0.68	0.68	0.68	10000
macro avg	0.68	0.68	0.67	10000
weighted avg	0.68	0.68	0.67	10000





CONVOLUTIONAL NEURAL NETWORKS – MODEL 4

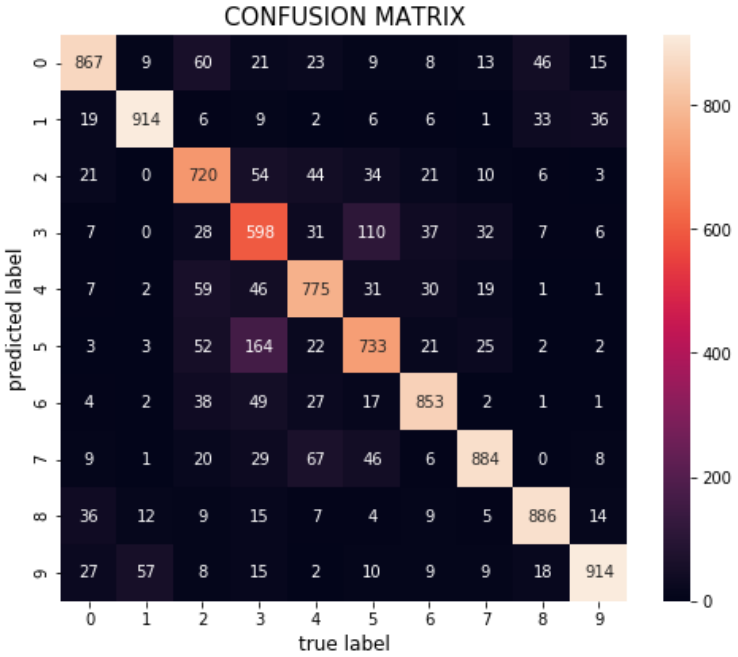
MODEL 4 (8 Layers)

Name: *model3*

- Variables: 3,072
- Convolutional layers: 8 +
- Max pooling layers: 4
- Dense Layers: 2
- Dropout layers: 5 (0.25 – 0.5) +
- Activation Functions used
  - Rectified Linear Unit (RELU)
  - Soft max
- Optimizer: SGD
- Loss = sparse categorical cross entropy
- Learning rate = 0.0001 -
- Batch size = 64 +
- Epochs = 100
- Total time: 05:28:11.04

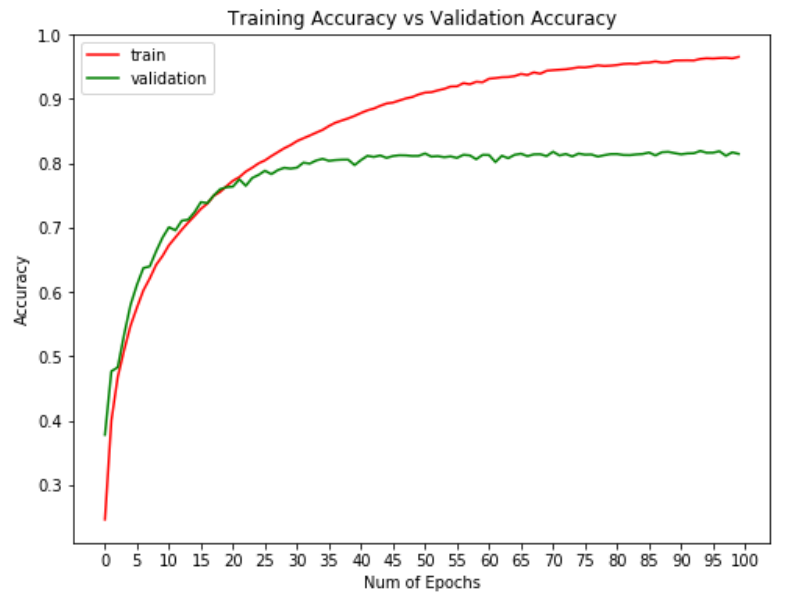
Test loss: 0.886

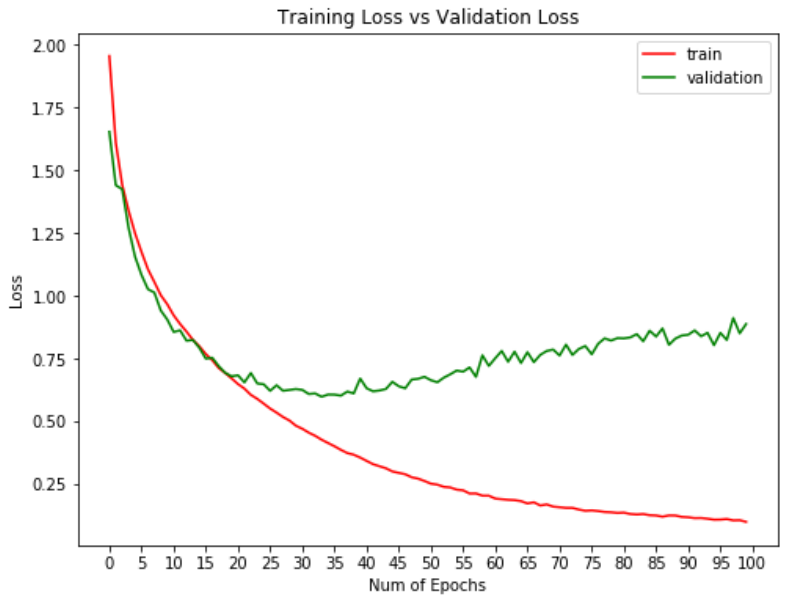
Test accuracy: 0.814



Classification Report

	precision	recall	f1-score	support
0	0.81	0.87	0.84	1000
1	0.89	0.91	0.90	1000
2	0.79	0.72	0.75	1000
3	0.70	0.60	0.64	1000
4	0.80	0.78	0.79	1000
5	0.71	0.73	0.72	1000
6	0.86	0.85	0.86	1000
7	0.83	0.88	0.85	1000
8	0.89	0.89	0.89	1000
9	0.86	0.91	0.88	1000
micro avg	0.81	0.81	0.81	10000
macro avg	0.81	0.81	0.81	10000
weighted avg	0.81	0.81	0.81	10000





CONVOLUTIONAL NEURAL NETWORKS – MODEL 5

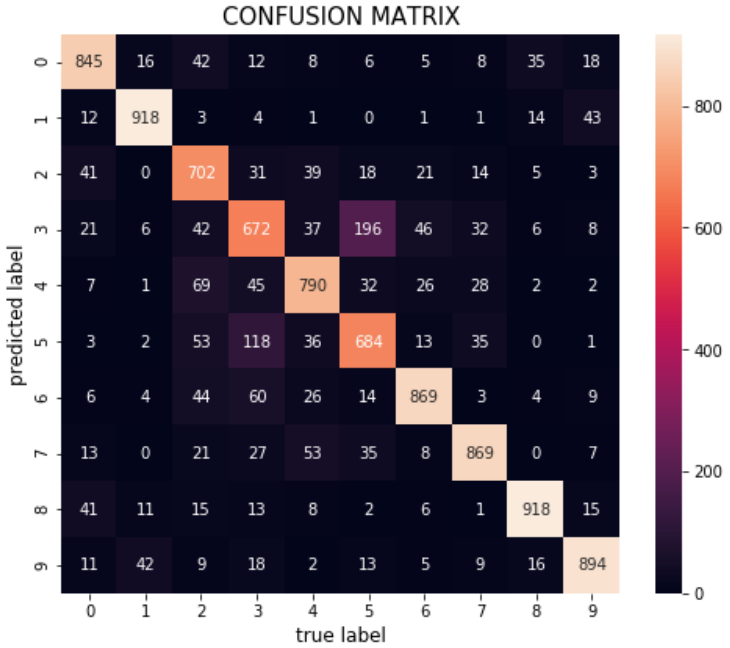
MODEL 5 (10 Layers)

Name: *model4*

- Variables: 3,072
- Convolutional layers: 10 +
- Max pooling layers: 4
- Dense Layers: 2
- Dropout layers: 5 (0.25 – 0.5) +
- Activation Functions used
  - Rectified Linear Unit (RELU)
  - Soft max
- Optimizer: SGD
- Loss = sparse categorical cross entropy
- Learning rate = 0.0001
- Batch size = 64
- Epochs = 100
- Total time: 10:30:20.44

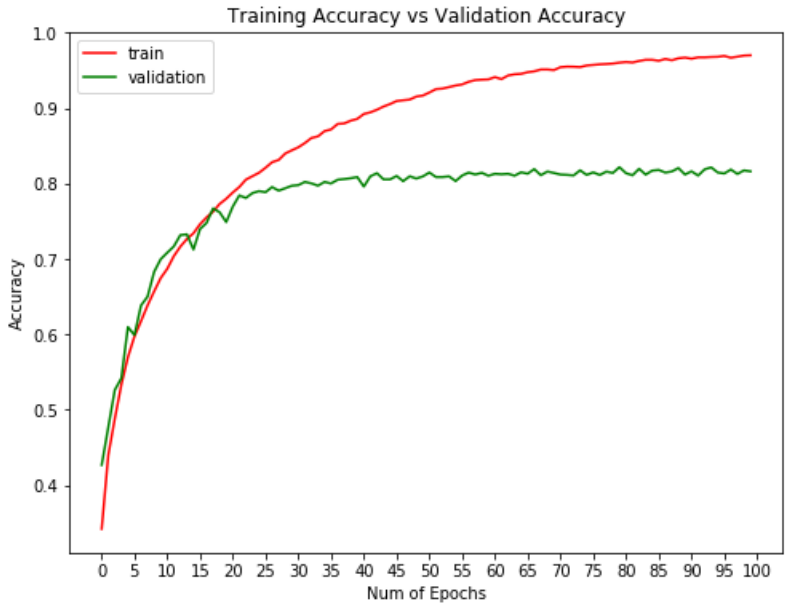
Test loss: 0.830

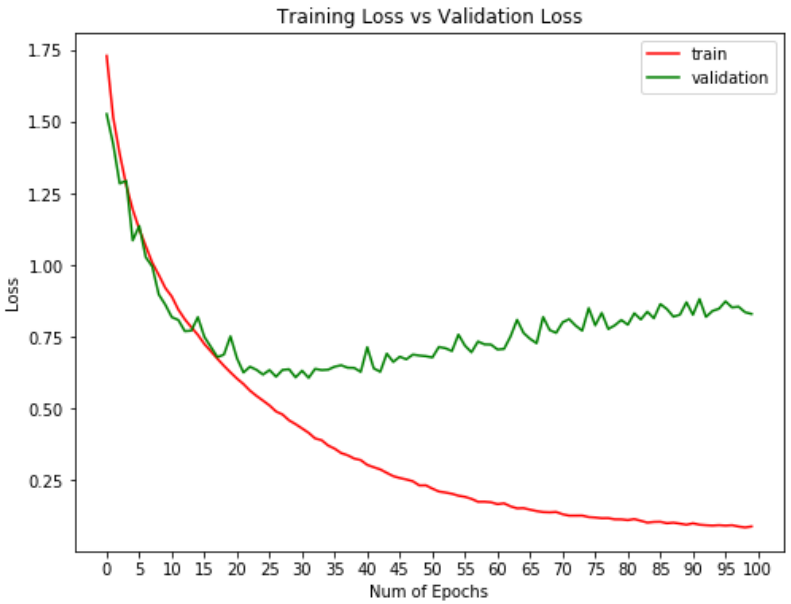
Test accuracy: 0.816



Classification Report

	precision	recall	f1-score	support
0	0.85	0.84	0.85	1000
1	0.92	0.92	0.92	1000
2	0.80	0.70	0.75	1000
3	0.63	0.67	0.65	1000
4	0.79	0.79	0.79	1000
5	0.72	0.68	0.70	1000
6	0.84	0.87	0.85	1000
7	0.84	0.87	0.85	1000
8	0.89	0.92	0.90	1000
9	0.88	0.89	0.89	1000
micro avg	0.82	0.82	0.82	10000
macro avg	0.82	0.82	0.82	10000
weighted avg	0.82	0.82	0.82	10000





MODEL COMPARISON

Traditional Models

	Random Forest	Logistic Regression	Gradient boost
Without PCA	R^2 Train Data: 0.64 R^2 Test Data: 0.43	R^2 Train Data: 0.52 R^2 Test Data: 0.36	R^2 Train Data: 0.94 R^2 Test Data: 0.44
Time	00:08:16.12	01:49:33.14	09:24:40.93
With PCA	R^2 Train Data: 0.60 R^2 Test Data: 0.43	R^2 Train Data: 0.38 R^2 Test Data: 0.38	R^2 Train Data: 0.92 R^2 Test Data: 0.44
Time	00:01:53.35	00:00:16.38	00:09:43.77

Convolutional Neural Networks

	Model1	Model1b	Model2b	Model3	Model4
Loss	1.72	1.25	0.89	0.886	0.830
Accuracy	0.38	0.55	0.68	0.814	0.816
Time	02:29:19.31	02:18:07.45	00:09:43.77	05:28:11.04	10:30:20.44

CONCLUSION

- Using Principle Component Analysis (PCA) was found effective for reducing dimensionality. It appeared to provide more time efficient results without compromising accuracy. In some instances results that were achieved using regular modeling techniques and which took extra long time to process were obtained in fraction of the time using PCA.

- Nevertheless traditional learning models such as Random Forest, Logistic Regression and Gradient boosting were time efficient however were unable to provide a higher accuracy in order for the model to be successful.

- For Convolutional Neural Networks adding more layers put heavy penalties on processing times. However at the same time CNN models displayed consistent results showing gradual fall in loss and increase in accuracy.

- In conclusion the models that serve the purpose well for image recognition include Convolutional Neural Networks and Random Forest. CNN although can be time consuming, however it offers far better accuracy than any other traditional machine learning models. Random Forest performed fairly well among traditional models however does not match up in delivering performance as that of Convolutional Neural Networks.

- For this project CNN (Model3) provides best possible results with 81% accuracy and a processing time of 5 hours and 30 minutes. This is followed by another CNN (Model4) which takes in additional layers and provides same level of accuracy within twice the amount of time taken by Model3 which is 10 hours 30 minutes.