CV ASSIGNMENT – 3

Name: Rahul Varma

Roll No: S20200010212

Dataset Used: CIFAR-10 Dataset

Three ML Model's Compared:

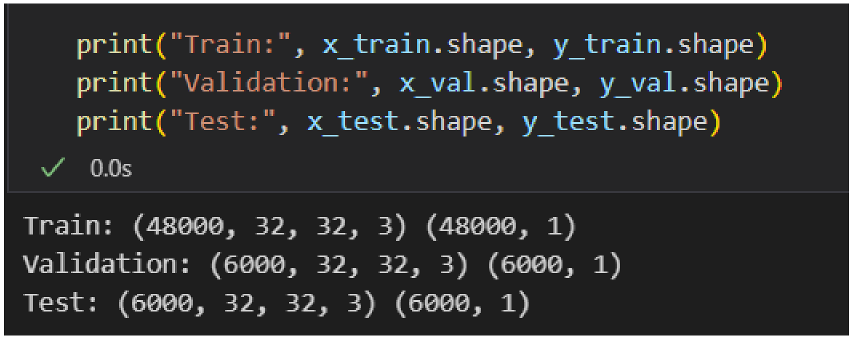
1) K-Nearest Neighbor
2) Non-Linear SVM
3) Deep Convolutional Neural Network (CNN)

Splitting Data:

1) 80% Train.
2) 10% Validation.
3) 10% Test.

as show in the image below.

Total Data = 60,000 -> 48,000 – 6,000 – 6,000.



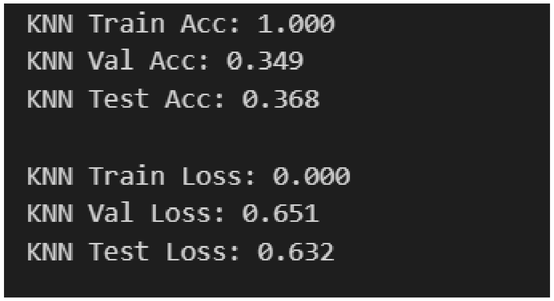
```
print("Train:", x_train.shape, y_train.shape)
print("Validation:", x_val.shape, y_val.shape)
print("Test:", x_test.shape, y_test.shape)
```

✓ 0.0s

```
Train: (48000, 32, 32, 3) (48000, 1)
Validation: (6000, 32, 32, 3) (6000, 1)
Test: (6000, 32, 32, 3) (6000, 1)
```

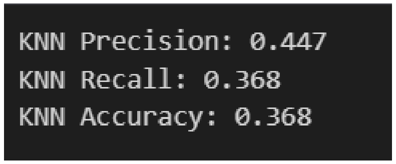
1)K-Nearest Neighbor:

Train, Validation, Test's Accuracy and Error (Loss) for KNN.



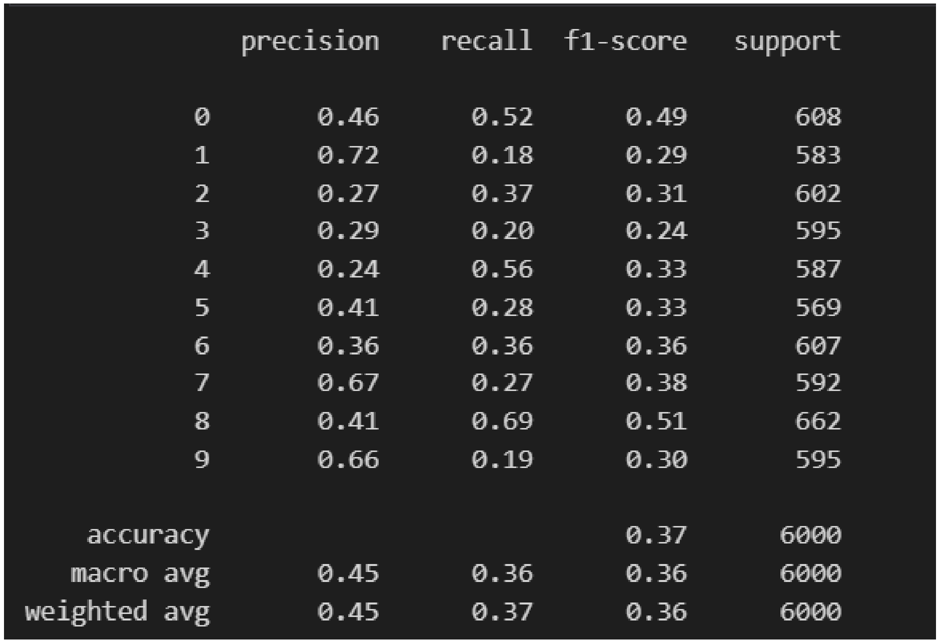
```
KNN Train Acc: 1.000
KNN Val Acc: 0.349
KNN Test Acc: 0.368

KNN Train Loss: 0.000
KNN Val Loss: 0.651
KNN Test Loss: 0.632
```

Precision, Recall and Accuracy for KNN:



```
KNN Precision: 0.447
KNN Recall: 0.368
KNN Accuracy: 0.368
```

Classification Report for KNN:



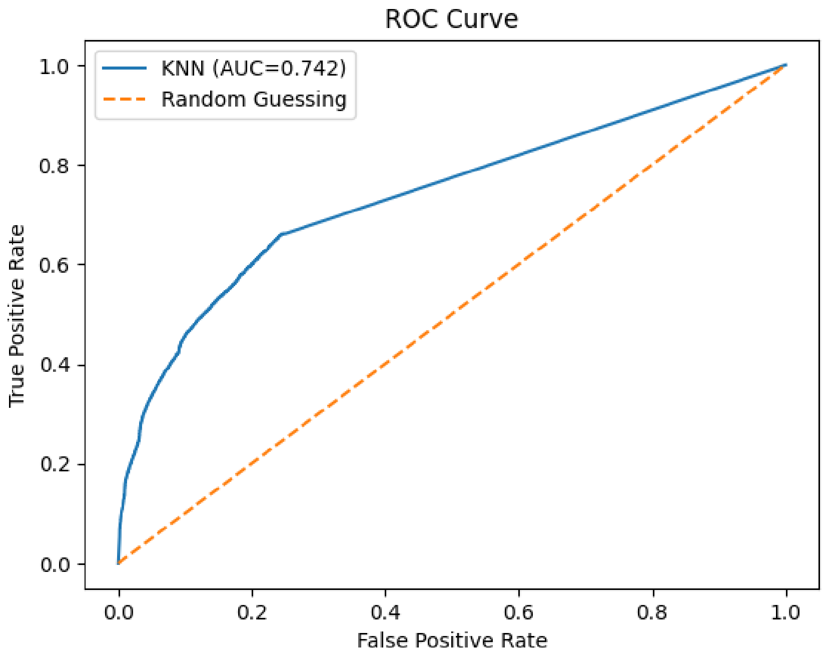
```
              precision    recall  f1-score   support

           0       0.46      0.52      0.49       608
           1       0.72      0.18      0.29       583
           2       0.27      0.37      0.31       602
           3       0.29      0.20      0.24       595
           4       0.24      0.56      0.33       587
           5       0.41      0.28      0.33       569
           6       0.36      0.36      0.36       607
           7       0.67      0.27      0.38       592
           8       0.41      0.69      0.51       662
           9       0.66      0.19      0.30       595

    accuracy                           0.37      6000
   macro avg       0.45      0.36      0.36      6000
weighted avg       0.45      0.37      0.36      6000
```

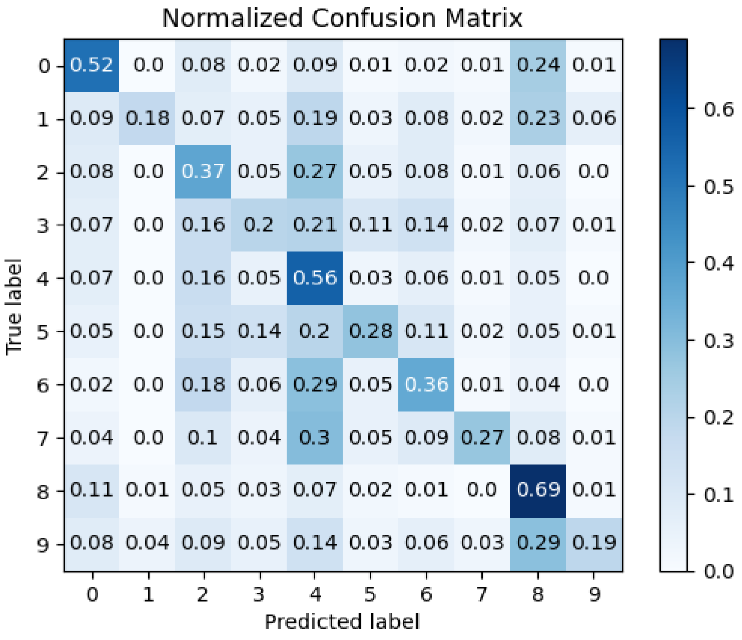
ROC Curve KNN Model:

False Positive Rate on X-axis and True Positive Rate on Y-axis.

And Confusion Matrix.

Area under the curve = 0.742

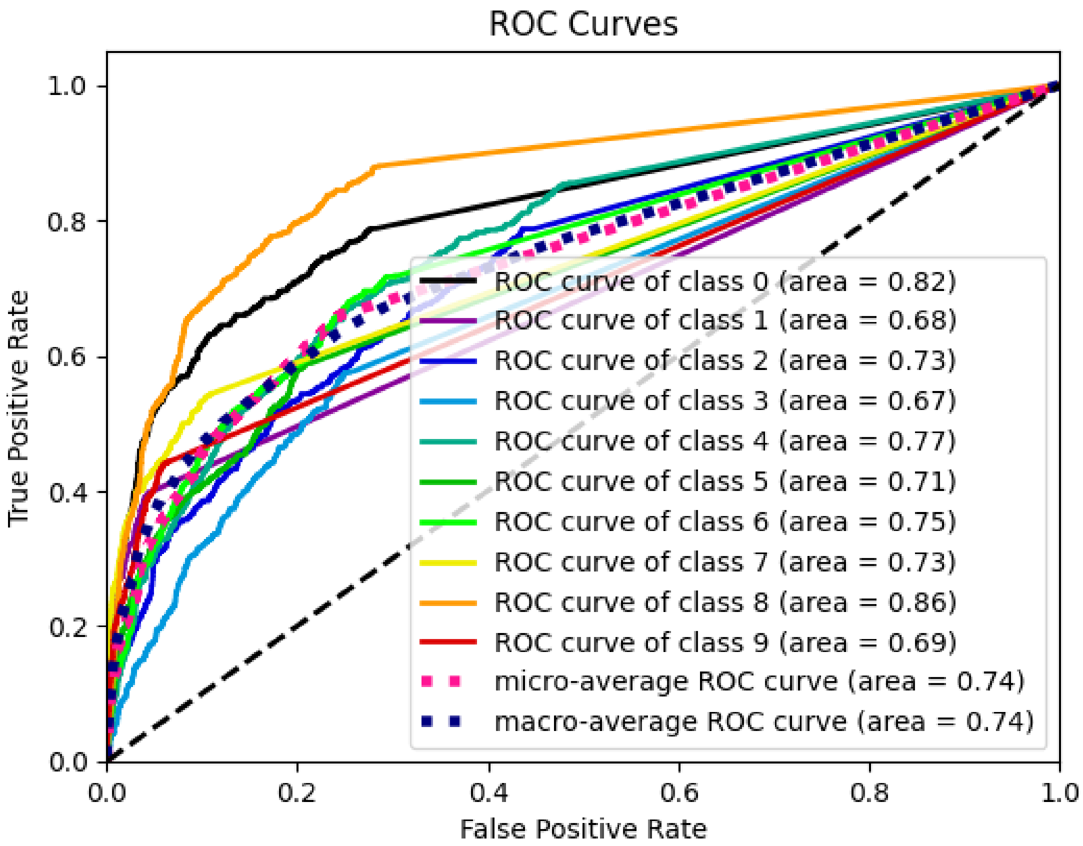




	0	1	2	3	4	5	6	7	8	9
0	0.52	0.0	0.08	0.02	0.09	0.01	0.02	0.01	0.24	0.01
1	0.09	0.18	0.07	0.05	0.19	0.03	0.08	0.02	0.23	0.06
2	0.08	0.0	0.37	0.05	0.27	0.05	0.08	0.01	0.06	0.0
3	0.07	0.0	0.16	0.2	0.21	0.11	0.14	0.02	0.07	0.01
4	0.07	0.0	0.16	0.05	0.56	0.03	0.06	0.01	0.05	0.0
5	0.05	0.0	0.15	0.14	0.2	0.28	0.11	0.02	0.05	0.01
6	0.02	0.0	0.18	0.06	0.29	0.05	0.36	0.01	0.04	0.0
7	0.04	0.0	0.1	0.04	0.3	0.05	0.09	0.27	0.08	0.01
8	0.11	0.01	0.05	0.03	0.07	0.02	0.01	0.0	0.69	0.01
9	0.08	0.04	0.09	0.05	0.14	0.03	0.06	0.03	0.29	0.19

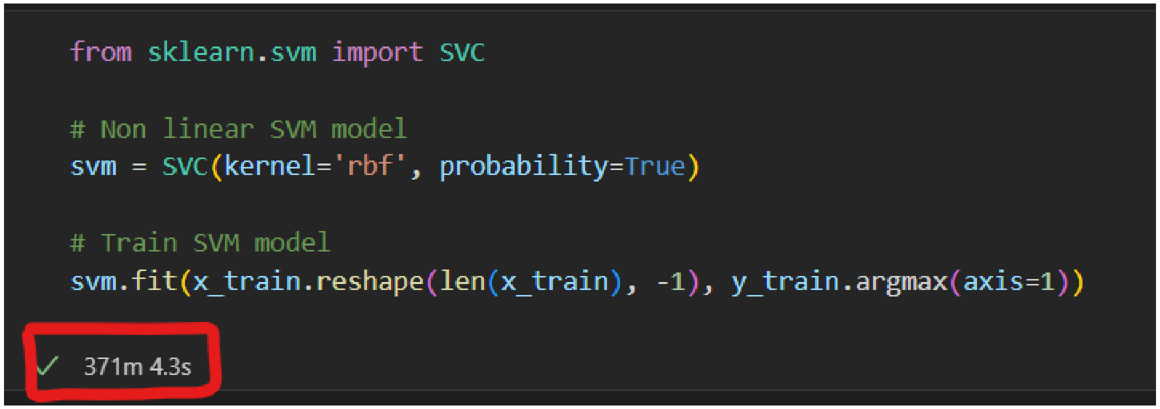
KNN

ROC Curve for each class in the CIFAR Data set:



2) Non-Linear SVM:

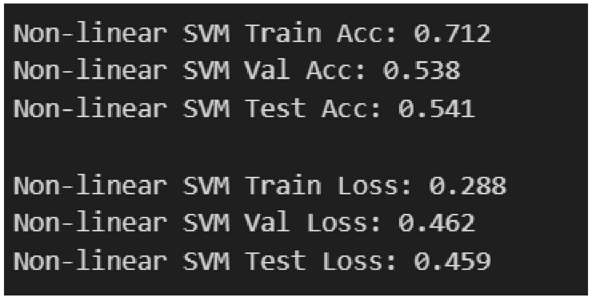
SVM Model took nearly 6 hour’s for Training.



```python
from sklearn.svm import SVC

# Non linear SVM model
svm = SVC(kernel='rbf', probability=True)

# Train SVM model
svm.fit(x_train.reshape(len(x_train), -1), y_train.argmax(axis=1))
```

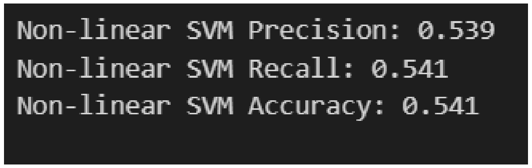
✓ 371m 4.3s

Train, Validation, Test’s Accuracy and Error (Loss) for SVM.



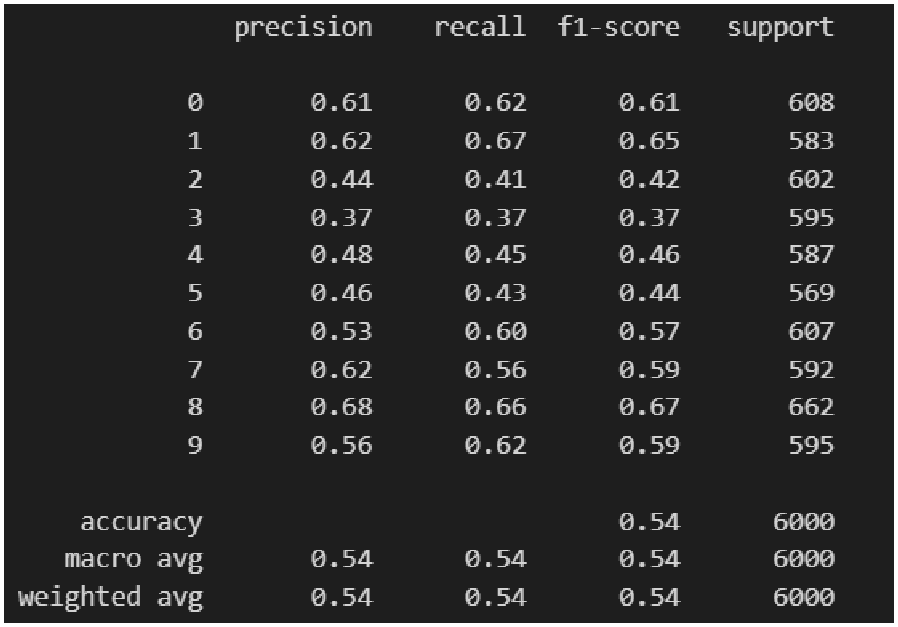
```
Non-linear SVM Train Acc: 0.712
Non-linear SVM Val Acc: 0.538
Non-linear SVM Test Acc: 0.541

Non-linear SVM Train Loss: 0.288
Non-linear SVM Val Loss: 0.462
Non-linear SVM Test Loss: 0.459
```

Precision, Recall and Accuracy for SVM.



```
Non-linear SVM Precision: 0.539
Non-linear SVM Recall: 0.541
Non-linear SVM Accuracy: 0.541
```

Classification Report for SVM:



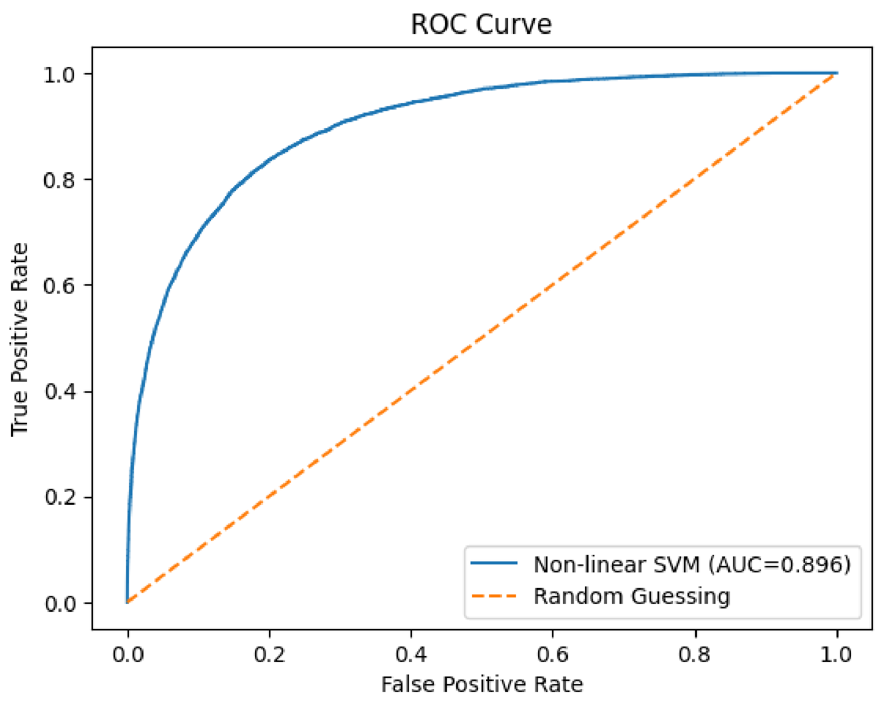
```
              precision    recall  f1-score   support

           0       0.61      0.62      0.61       608
           1       0.62      0.67      0.65       583
           2       0.44      0.41      0.42       602
           3       0.37      0.37      0.37       595
           4       0.48      0.45      0.46       587
           5       0.46      0.43      0.44       569
           6       0.53      0.60      0.57       607
           7       0.62      0.56      0.59       592
           8       0.68      0.66      0.67       662
           9       0.56      0.62      0.59       595

    accuracy                           0.54      6000
   macro avg       0.54      0.54      0.54      6000
weighted avg       0.54      0.54      0.54      6000
```

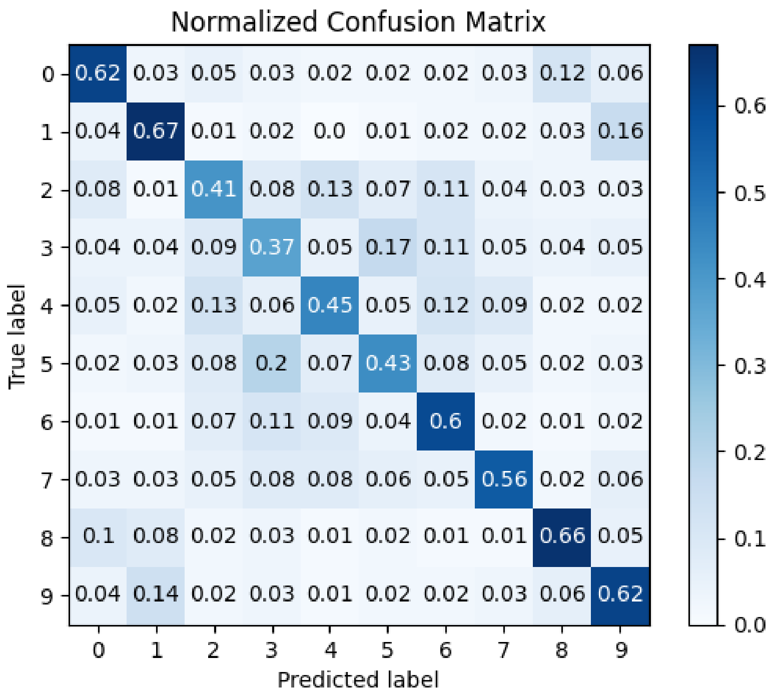
ROC Curve SVM Model:

False Positive Rate on X-axis and True Positive Rate on Y-axis.

And Confusion Matrix.

Area Under the Curve is 0.896

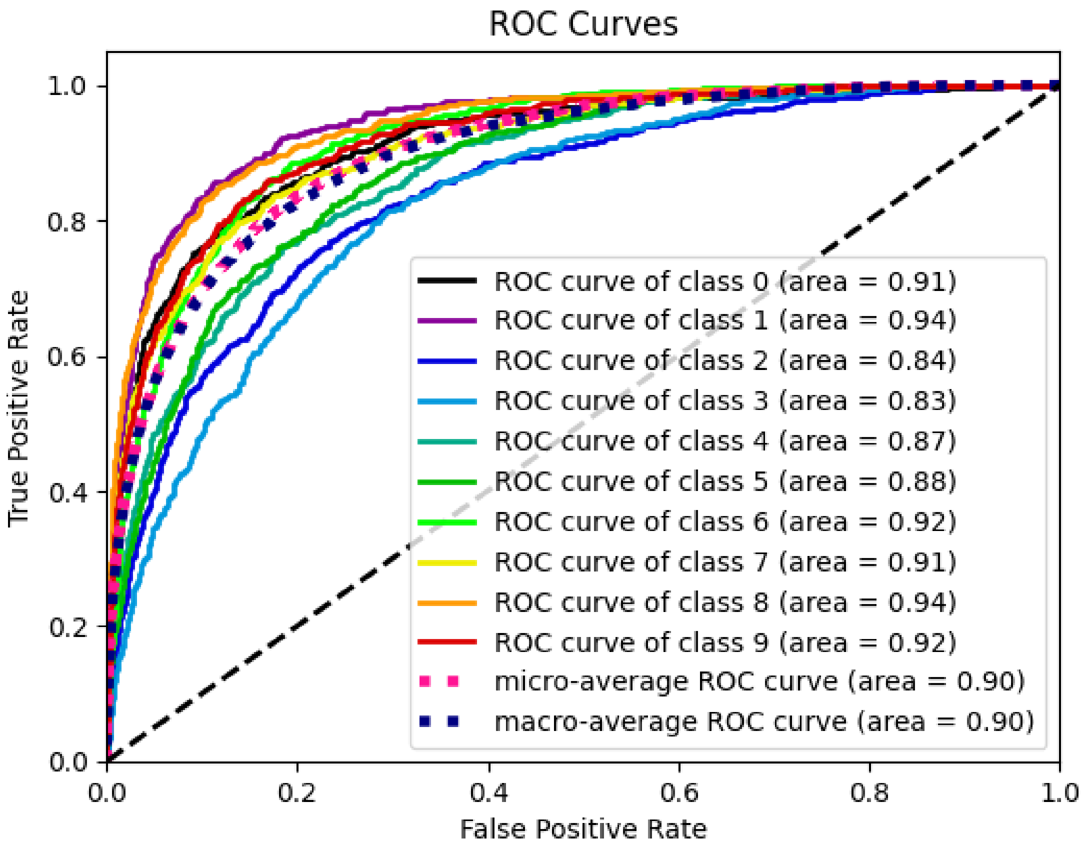




	0	1	2	3	4	5	6	7	8	9
0	0.62	0.03	0.05	0.03	0.02	0.02	0.02	0.03	0.12	0.06
1	0.04	0.67	0.01	0.02	0.0	0.01	0.02	0.02	0.03	0.16
2	0.08	0.01	0.41	0.08	0.13	0.07	0.11	0.04	0.03	0.03
3	0.04	0.04	0.09	0.37	0.05	0.17	0.11	0.05	0.04	0.05
4	0.05	0.02	0.13	0.06	0.45	0.05	0.12	0.09	0.02	0.02
5	0.02	0.03	0.08	0.2	0.07	0.43	0.08	0.05	0.02	0.03
6	0.01	0.01	0.07	0.11	0.09	0.04	0.6	0.02	0.01	0.02
7	0.03	0.03	0.05	0.08	0.08	0.06	0.05	0.56	0.02	0.06
8	0.1	0.08	0.02	0.03	0.01	0.02	0.01	0.01	0.66	0.05
9	0.04	0.14	0.02	0.03	0.01	0.02	0.02	0.03	0.06	0.62

SVM

ROC Curve for each class in the CIFAR Data set:



3)CNN Model:

Trained the Model for 20 Epochs.

CNN model that includes a pre-trained VGG16 model as a feature extractor. The model has two sets of convolutional layers, pooling layers, and dropout layers, followed by a fully connected dense layer and a softmax output layer.

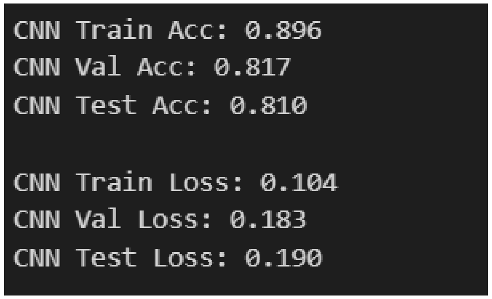
The layers used in the model are:

- VGG16 model
- Conv2D layer
- MaxPooling2D layer
- Dropout layer
- Flatten layer
- Dense layer
- Softmax layer

The VGG16 model is used as the initial layers of the model, followed by two sets of Conv2D, MaxPooling2D, and Dropout layers. The output of these layers is flattened and passed through a dense layer with ReLU activation and another Dropout layer. Finally, the output is passed through a dense layer with softmax activation to produce the final class probabilities.

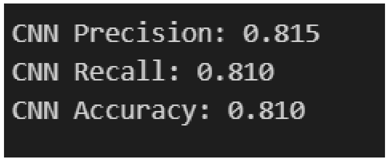
The model is compiled with categorical cross-entropy loss, Adam optimizer, and accuracy metrics. It is trained for 20 epochs on the training data with a batch size of 128 and validated on a separate validation set.

CNN Train, Validation, Test’s Accuracy and Error(Loss) :



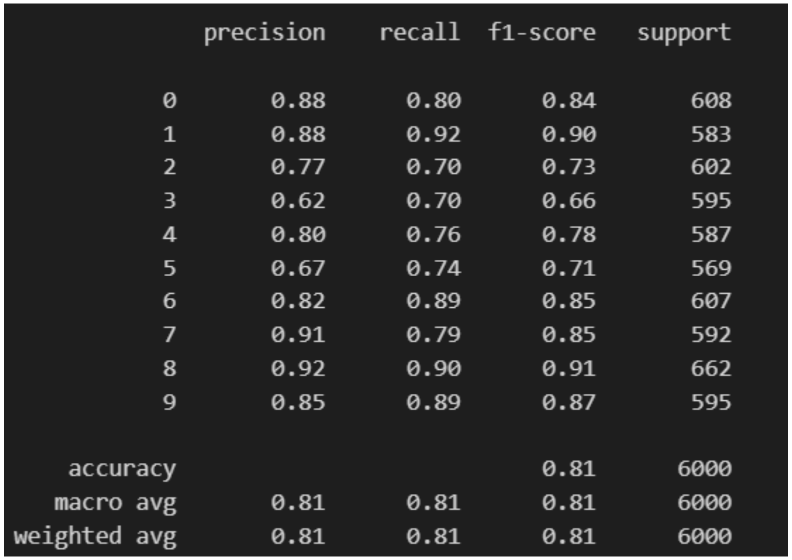
```
CNN Train Acc: 0.896
CNN Val Acc: 0.817
CNN Test Acc: 0.810

CNN Train Loss: 0.104
CNN Val Loss: 0.183
CNN Test Loss: 0.190
```

Precision, Recall and Accuracy for CNN.



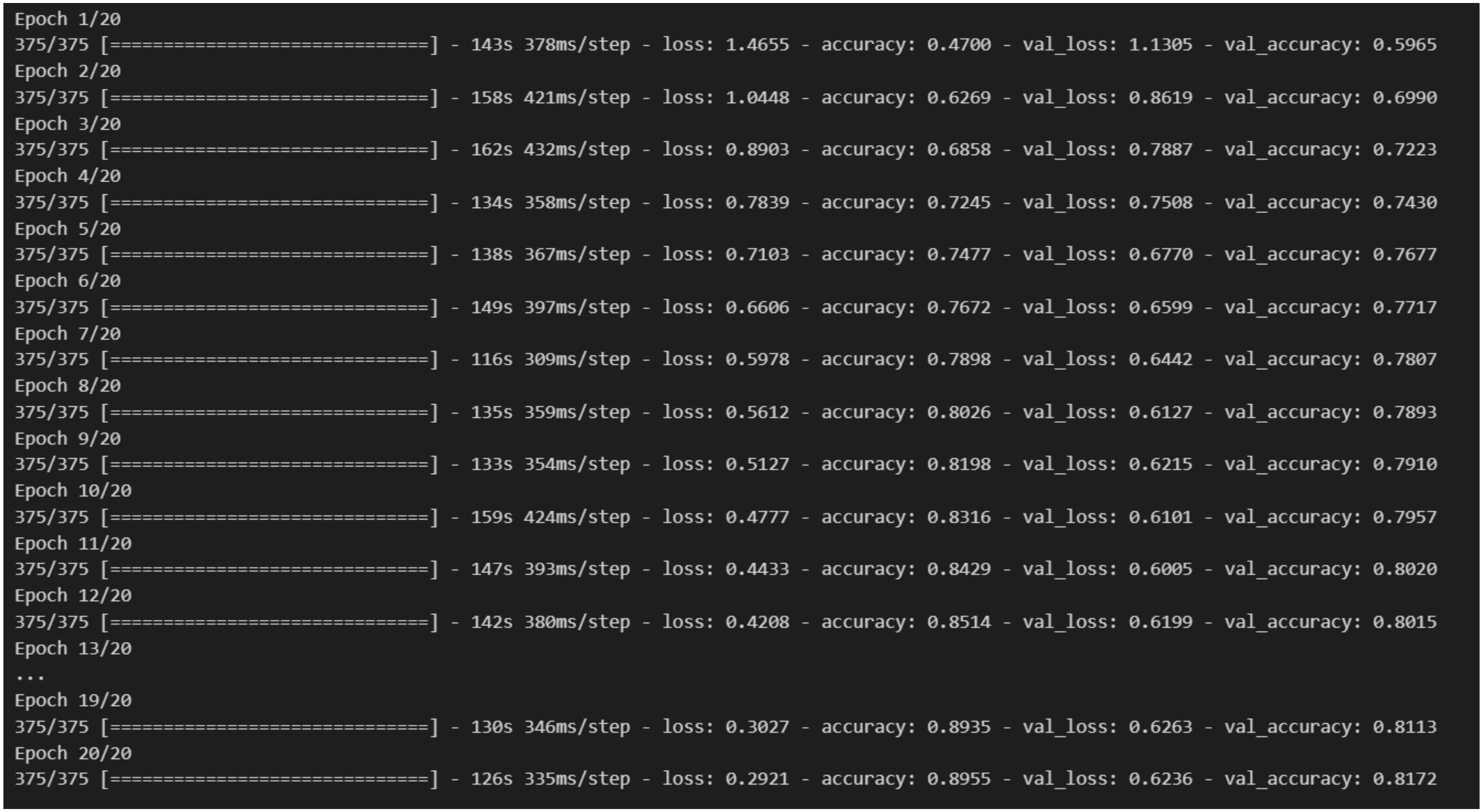
```
CNN Precision: 0.815
CNN Recall: 0.810
CNN Accuracy: 0.810
```

Classification Report for CNN:



```
              precision    recall  f1-score   support

           0       0.88      0.80      0.84       608
           1       0.88      0.92      0.90       583
           2       0.77      0.70      0.73       602
           3       0.62      0.70      0.66       595
           4       0.80      0.76      0.78       587
           5       0.67      0.74      0.71       569
           6       0.82      0.89      0.85       607
           7       0.91      0.79      0.85       592
           8       0.92      0.90      0.91       662
           9       0.85      0.89      0.87       595

    accuracy                           0.81      6000
   macro avg       0.81      0.81      0.81      6000
weighted avg       0.81      0.81      0.81      6000
```

Training for 20 Epochs:

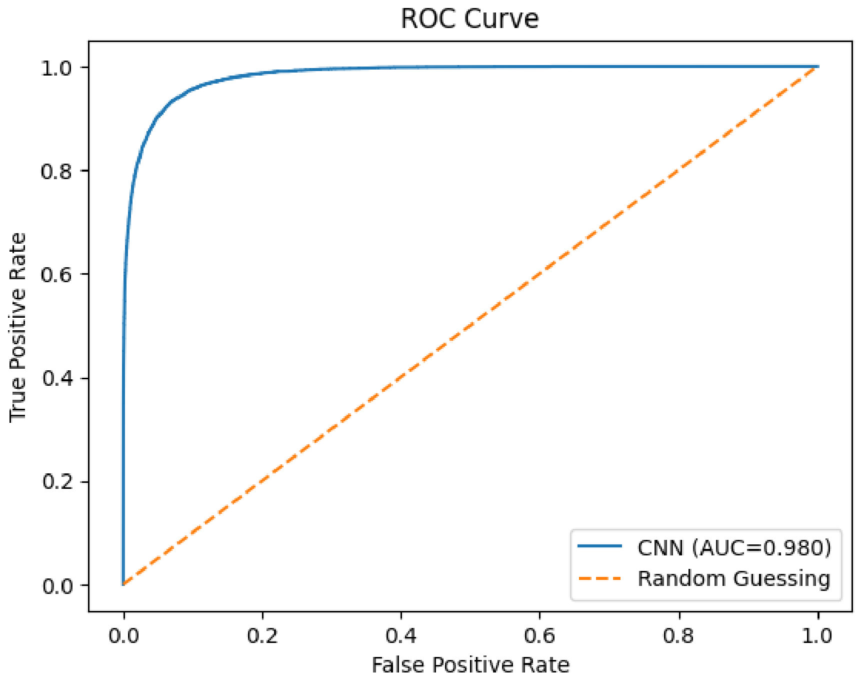


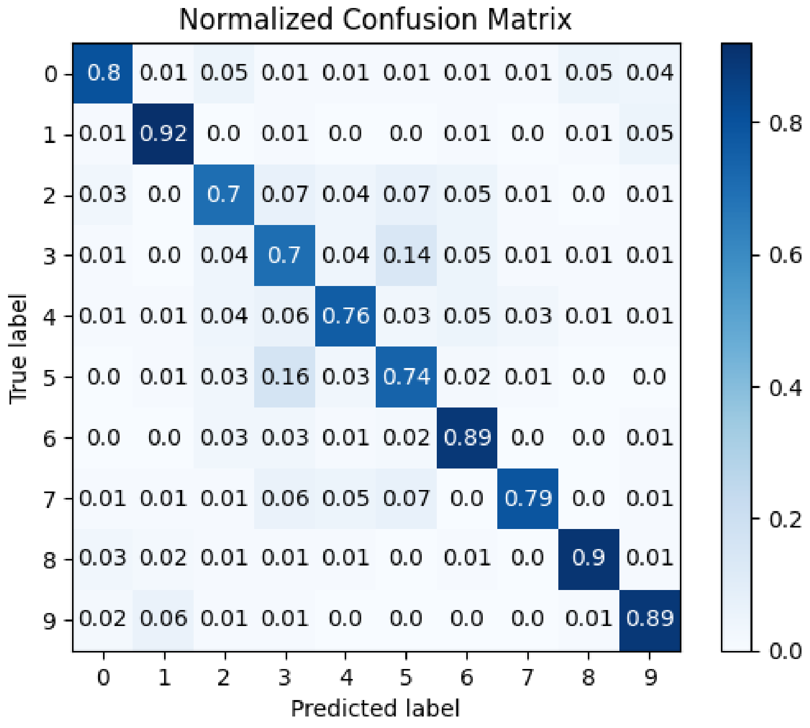
```
Epoch 1/20
375/375 [==============================] - 143s 378ms/step - loss: 1.4655 - accuracy: 0.4700 - val_loss: 1.1305 - val_accuracy: 0.5965
Epoch 2/20
375/375 [==============================] - 158s 421ms/step - loss: 1.0448 - accuracy: 0.6269 - val_loss: 0.8619 - val_accuracy: 0.6990
Epoch 3/20
375/375 [==============================] - 162s 432ms/step - loss: 0.8903 - accuracy: 0.6858 - val_loss: 0.7887 - val_accuracy: 0.7223
Epoch 4/20
375/375 [==============================] - 134s 358ms/step - loss: 0.7839 - accuracy: 0.7245 - val_loss: 0.7508 - val_accuracy: 0.7430
Epoch 5/20
375/375 [==============================] - 138s 367ms/step - loss: 0.7103 - accuracy: 0.7477 - val_loss: 0.6770 - val_accuracy: 0.7677
Epoch 6/20
375/375 [==============================] - 149s 397ms/step - loss: 0.6606 - accuracy: 0.7672 - val_loss: 0.6599 - val_accuracy: 0.7717
Epoch 7/20
375/375 [==============================] - 116s 309ms/step - loss: 0.5978 - accuracy: 0.7898 - val_loss: 0.6442 - val_accuracy: 0.7807
Epoch 8/20
375/375 [==============================] - 135s 359ms/step - loss: 0.5612 - accuracy: 0.8026 - val_loss: 0.6127 - val_accuracy: 0.7893
Epoch 9/20
375/375 [==============================] - 133s 354ms/step - loss: 0.5127 - accuracy: 0.8198 - val_loss: 0.6215 - val_accuracy: 0.7910
Epoch 10/20
375/375 [==============================] - 159s 424ms/step - loss: 0.4777 - accuracy: 0.8316 - val_loss: 0.6101 - val_accuracy: 0.7957
Epoch 11/20
375/375 [==============================] - 147s 393ms/step - loss: 0.4433 - accuracy: 0.8429 - val_loss: 0.6005 - val_accuracy: 0.8020
Epoch 12/20
375/375 [==============================] - 142s 380ms/step - loss: 0.4208 - accuracy: 0.8514 - val_loss: 0.6199 - val_accuracy: 0.8015
Epoch 13/20
...
Epoch 19/20
375/375 [==============================] - 130s 346ms/step - loss: 0.3027 - accuracy: 0.8935 - val_loss: 0.6263 - val_accuracy: 0.8113
Epoch 20/20
375/375 [==============================] - 126s 335ms/step - loss: 0.2921 - accuracy: 0.8955 - val_loss: 0.6236 - val_accuracy: 0.8172
```

ROC Curve CNN Model:

False Positive Rate on X-axis and True Positive Rate on Y-axis. And Confusion Matrix.

Area Under the Curve is 0.980

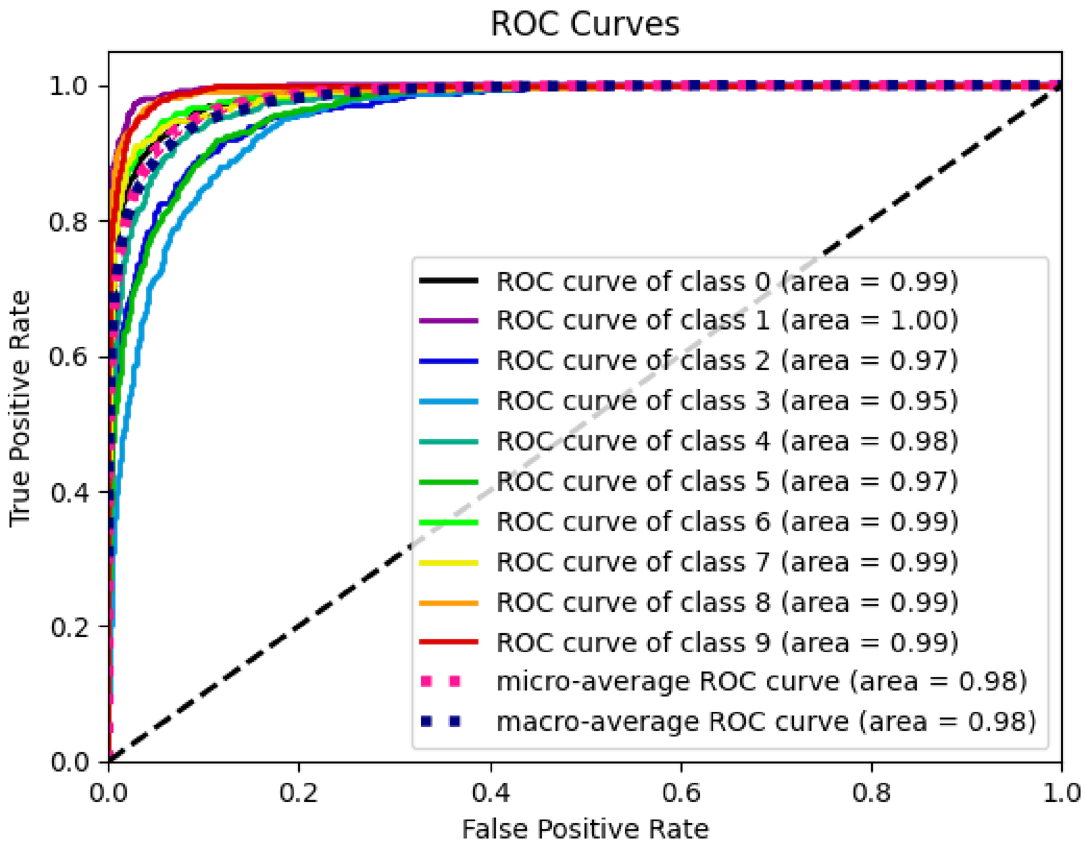

ROC Curve


Normalized Confusion Matrix

True label \ Predicted label	0	1	2	3	4	5	6	7	8	9
0	0.8	0.01	0.05	0.01	0.01	0.01	0.01	0.01	0.05	0.04
1	0.01	0.92	0.0	0.01	0.0	0.0	0.01	0.0	0.01	0.05
2	0.03	0.0	0.7	0.07	0.04	0.07	0.05	0.01	0.0	0.01
3	0.01	0.0	0.04	0.7	0.04	0.14	0.05	0.01	0.01	0.01
4	0.01	0.01	0.04	0.06	0.76	0.03	0.05	0.03	0.01	0.01
5	0.0	0.01	0.03	0.16	0.03	0.74	0.02	0.01	0.0	0.0
6	0.0	0.0	0.03	0.03	0.01	0.02	0.89	0.0	0.0	0.01
7	0.01	0.01	0.01	0.06	0.05	0.07	0.0	0.79	0.0	0.01
8	0.03	0.02	0.01	0.01	0.01	0.0	0.01	0.0	0.9	0.01
9	0.02	0.06	0.01	0.01	0.0	0.0	0.0	0.0	0.01	0.89

CNN

ROC Curve for each class in the CIFAR Data set:



Final Report from the above observation on ROC Analysis:

Based on the ROC curve analysis, the CNN model using VGGNet has the highest AUC score of 0.98, indicating that it has the best overall performance for distinguishing between the classes. The non-linear SVM model comes in second with an AUC score of 0.896, while KNN has the lowest AUC score of 0.742.

In terms of accuracy, the CNN model also has the highest performance on both the validation and test sets, achieving 81.7% and 81.0% accuracy respectively. The non-linear SVM model has the second-best accuracy, achieving 53.8% on the validation set and 54.1% on the test set. KNN has the lowest accuracy on both sets, achieving only 34.9% on the validation set and 36.8% on the test set.

The CNN model also has the lowest loss on both the validation and test sets, indicating better performance in minimizing the error. Additionally, the CNN model has the highest precision and recall values, indicating that it has the best overall performance for identifying true positive cases.

In conclusion, based on the ROC curve analysis, accuracy, loss, precision, and recall values, the CNN model using VGGNet is the best among the given models for this classification task.

Based on the given results, we can conclude that the CNN model using VGGNet architecture is the best among the three models for the CIFAR dataset. This conclusion is based on the analysis of ROC curves, where the CNN model has the highest AUC value of 0.980 compared to KNN (0.742) and Non-linear SVM (0.896).

Furthermore, the CNN model has higher accuracy and lower loss values for both training and validation sets compared to KNN and Non-linear SVM. In terms of precision, recall, and accuracy, the CNN model also outperforms the other two models.

Therefore, we can conclude that the CNN model using VGGNet architecture is the most suitable and effective model for the CIFAR dataset.