# ENEE 633 Project- 2

# Report

## Problem Statement

The aim of this project is to implement different conventional classifiers to achieve hand-written digits recognition, as well as performing transfer learning in an image classification task. The project is divided into two parts. The subtasks of each part are mentioned below.

**Part 1: Handwritten Digit Recognition**

1. SVM
   1.1. Perform Multiple Discriminant Analysis (MDA) followed by Support Vector Machine Classifier using 3 types of Kernels- Linear, Polynomial and RBF
   1.2. Perform Principal Component Analysis (PCA) followed by Support Vector Machine Classifier using 3 types of Kernels- Linear, Polynomial and RBF

2. Logistic Regression
   2.1. Perform Multiple Discriminant Analysis (MDA) followed by multinomial Logistic Regression
   2.2. Perform Principal Component Analysis (PCA) followed by multinomial Logistic Regression

3. Deep Learning
   3.1. Build a convolutional neural network for classification

<u>Dataset:</u> MNIST Dataset

**Part 2: Transfer Learning**

1. Train a simple Convolutional neural network on the dataset
2. Use the transfer learning technique to train the dataset

<u>Dataset:</u> 10 Monkey species dataset

## Implementation

**Part 1:**

MNIST dataset consists of a training set with 60000, 28 x 28 grayscale images of handwritten digits (10 classes) and a testing set with 10000 images.

The MNIST dataset was imported from keras for the project. The three tasks of SVM, Logistic regression and deep learning were implemented on this model. PCA and MDA were used to reduce the dimensionality. The images were preprocessed and normalized before implementing the classification.

**Part 2:**

The 10 Monkey species dataset consists of a training set with 1098 jpeg images labeled as n0~n9, each corresponding a monkey species. The validation dataset consists of 272 images labeled as n0~n9, each corresponding a monkey species. All the images are 400x300 px or larger.

The dataset was downloaded from Kaggle and used in the dataset using the directory id. First a simple CNN was implemented followed by Transfer learning which was done using the VGG16 pretrained weights. VGG16 weights were imported from keras. The improvement in accuracy was observed on using VGG16 pretrained model.

## Results

**Part 1: Handwritten Digit Recognition**

1. SVM
    1.1. Perform Multiple Discriminant Analysis (MDA) followed by Support Vector Machine Classifier using 3 types of Kernels- Linear, Polynomial and RBF

| Type | Linear Kernel | RBF Kernel | Polynomial Kernel(degree=3) | Average |
|---|---|---|---|---|
| Testing accuracy | 0.89 | 0.92 | 0.88 | 0.89 |

1.2. Perform Principal Component Analysis (PCA) followed by Support Vector Machine Classifier using 3 types of Kernels- Linear, Polynomial and RBF

| Type | Linear Kernel | RBF Kernel | Polynomial Kernel(degree=3) | Average |
|---|---|---|---|---|
| Testing accuracy | 0.94 | 0.97 | 0.97 | 0.96 |

```
For linear classifier
[[ 945    0    5    2    0    9   13    1    5
 [   0 1095    4    6    1    0    3    2   24
 [  13   11  904   25   12    6   22    9   28
 [   3    2   30  877    1   45    2   16   27
 [   0    2    6    2  910    1   16    5    8
 [  13    2   10   50   11  743   15    9   34
 [  18    2   12    0    9   20  894    0    3
 [   2   17   20   12   11    1    0  908    3
 [  11   23   10   31   17   48   16   10  796
 [   9    2    3   15   61    7    1   41    9
              precision    recall  f1-score   s

           0       0.93      0.96      0.95
           1       0.95      0.96      0.96
           2       0.90      0.88      0.89
           3       0.86      0.87      0.86
           4       0.88      0.93      0.90
           5       0.84      0.83      0.84
           6       0.91      0.93      0.92
           7       0.91      0.88      0.90
           8       0.85      0.82      0.83
           9       0.88      0.85      0.87

    accuracy                           0.89
   macro avg       0.89      0.89      0.89
weighted avg       0.89      0.89      0.89
```

Fig 1.1.1- Accuracy of linear kernel svm classifier using MDA

```
For linear classifier
[[ 959    0    3    3    1    7    5    0    2
 [   0 1121    6    2    0    1    2    1    2
 [   8    7  967   12    6    1    6    7   17
 [   3    0   12  948    1   16    2    7   19
 [   3    1    8    3  932    0    4    6    5
 [   7    7    4   29    8  808    8    1   18
 [  11    3   12    1    9   13  908    0    1
 [   1    5   24   11    9    2    0  954    3
 [   6    6   12   24    7   22    6    7  883
 [   4    6    4   10   27    4    0   21    8
              precision    recall  f1-score   su

           0       0.96      0.98      0.97
           1       0.97      0.99      0.98
           2       0.92      0.94      0.93
           3       0.91      0.94      0.92
           4       0.93      0.95      0.94
           5       0.92      0.91      0.92
           6       0.96      0.95      0.96
           7       0.95      0.93      0.94
           8       0.92      0.91      0.91
           9       0.95      0.92      0.93

    accuracy                           0.94
   macro avg       0.94      0.94      0.94
weighted avg       0.94      0.94      0.94
```

Fig 1.2.1- Accuracy of linear kernel svm classifier using PCA

2. Logistic Regression

2.1. Perform Multiple Discriminant Analysis (MDA) followed by multinomial Logistic Regression

| Testing accuracy | 0.89 |
|---|---|

2.2. Perform Principal Component Analysis (PCA) followed by multinomial Logistic Regression

| | |
|---|---|
| Testing accuracy | 0.9 |

```
[[ 948    0    3    2    0   12    8    2    5    0]
 [   0 1092    5    6    1    0    3    2   26    0]
 [  12   12  887   26   12    5   23   17   30    8]
 [   3    1   26  876    3   45    1   17   31    7]
 [   1   10    7    0  891    1   15    4    9   44]
 [   9    2    6   45   16  737   15   16   42    4]
 [  18    4   12    0   12   22  883    2    4    1]
 [   3   18   18   14   10    1    0  902    1   61]
 [  11   21   11   30   19   46   18    7  799   12]
 [  12    3    3   12   60    7    1   46   15  850]]
              precision    recall  f1-score   support

           0       0.93      0.97      0.95       980
           1       0.94      0.96      0.95      1135
           2       0.91      0.86      0.88      1032
           3       0.87      0.87      0.87      1010
           4       0.87      0.91      0.89       982
           5       0.84      0.83      0.83       892
           6       0.91      0.92      0.92       958
           7       0.89      0.88      0.88      1028
           8       0.83      0.82      0.83       974
           9       0.86      0.84      0.85      1009

    accuracy                           0.89     10000
   macro avg       0.88      0.89      0.88     10000
weighted avg       0.89      0.89      0.89     10000
```

Fig 2.1.1- Accuracy of (MDA) followed by multinomial Logistic Regression

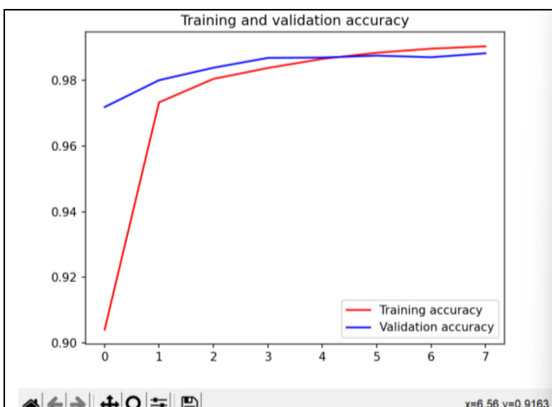
```
[[ 948    0    1    2    0   14   10    2    3    0]
 [   0 1094    4    7    1    4    4    0   21    0]
 [  16   12  884   23   16    2   15   24   37    3]
 [   6    4   20  889    1   36    6   18   22    8]
 [   2    6    4    0  917    2   13    3    3   32]
 [  14    5    6   40   19  737   18   16   27   10]
 [  15    5   10    0   13   18  892    1    4    0]
 [   1   21   21    6   12    0    1  924    1   41]
 [  10   20    9   30   21   38   10   18  802   16]
 [  16    7    5   12   55    9    0   34    4  867]]
              precision    recall  f1-score   support

           0       0.92      0.97      0.94       980
           1       0.93      0.96      0.95      1135
           2       0.92      0.86      0.89      1032
           3       0.88      0.88      0.88      1010
           4       0.87      0.93      0.90       982
           5       0.86      0.83      0.84       892
           6       0.92      0.93      0.93       958
           7       0.89      0.90      0.89      1028
           8       0.87      0.82      0.85       974
           9       0.89      0.86      0.87      1009

    accuracy                           0.90     10000
   macro avg       0.89      0.89      0.89     10000
weighted avg       0.90      0.90      0.89     10000
```

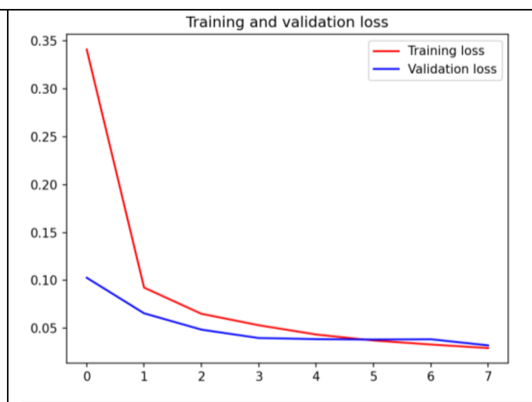
Fig 2.2.1- Accuracy of (PCA) followed by multinomial Logistic Regression

3. Deep Learning

3.1. Build a convolutional neural network for classification

| | |
|---|---|
| Training accuracy | 0.98 |
| Testing accuracy | 0.98 |

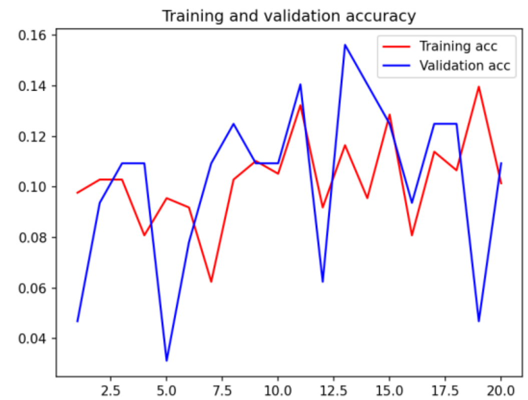


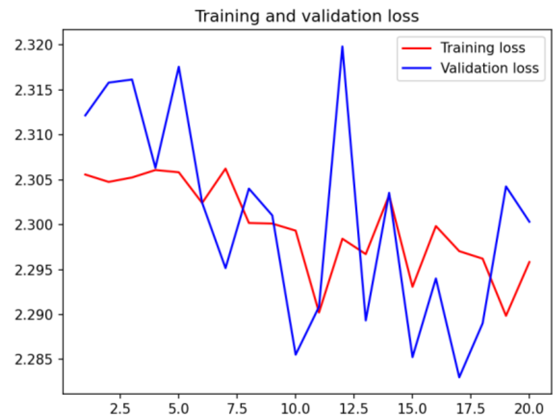



**Part 2: Transfer Learning**

1. Train a simple Convolutional neural network on the dataset

| | |
|---|---|
| Testing accuracy | 0.1094 |

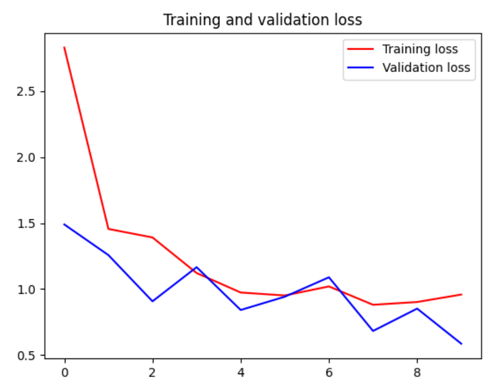


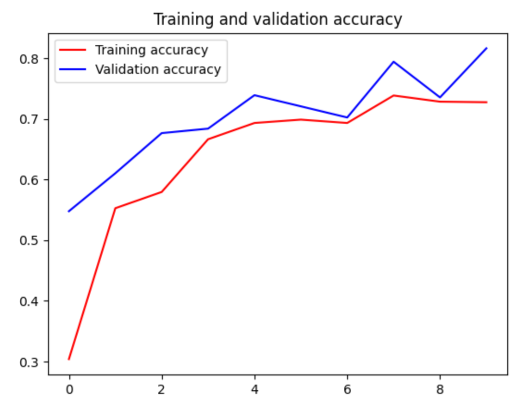




2. Use the transfer learning technique to train the dataset

   The model used for transfer learning was VGG16.

| | |
|---|---|
| Testing accuracy | 0.78 |




# Discussion

## Part 1

1. The kernel SVM was implemented using three kernels- linear, rbf and polynomial with degree 3. The highest accuracy was obtained for the rbf kernel when multiple discriminant analysis was applied to the dataset first. When principal component analysis was applied to the data first, the maximum accuracy was obtained for both rbf and polynomial kernel of degree 3. When comparing with respect to dimensionality reduction techniques, PCA had a better performance compared to MDA.
2. For the multinomial logistic regression classifier, in the cases where both PCA and MDA were used, PCA followed by regression had better accuracy of 0.90.
3. In the case of using CNN, the built model gave an accuracy of 0.98.

In general, the highest accuracy for classification of hand written digits were obtained using Deep learning.

**Part 2**

For the 10-monkey classification dataset, a simple, 3-5 neural network was implemented first which gave the accuracy of 0.1094. The same dataset was then classified using the VGG16 model with pretrained weights and a considerable improvement in accuracy was achieved. The new accuracy was 0.78.

## Conclusion

For the hand written digit classification, the Computational neural network was the best model for classification in terms of accuracy. However, it may take a longer time to classify using a CNN compared to SVM or logistic regression.

A huge improvement was observed for the 10-monkey classification problem owing to the use of pretrained VGG16 model for transfer learning. The pre-learned weights help in improving the learning of the new problem.