COL774 Assignment 4

1.

a)

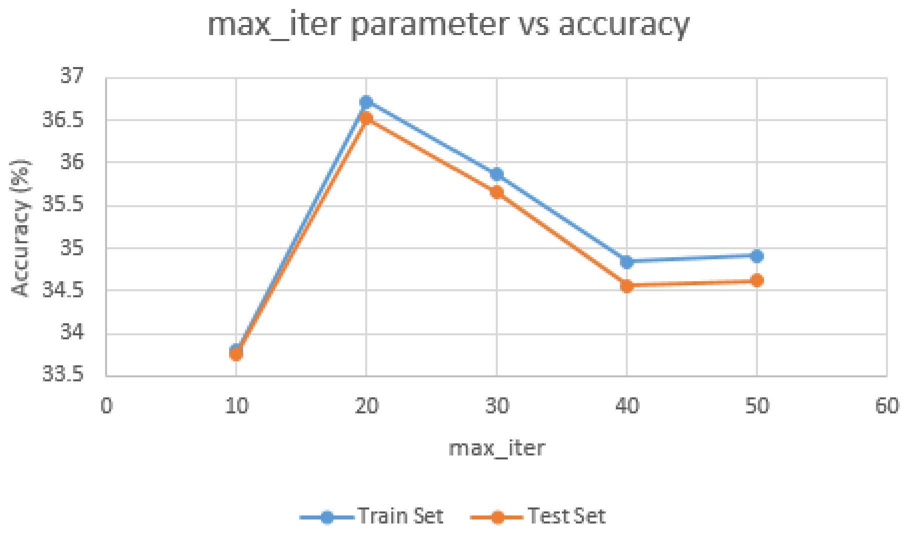
(ii)

Training accuracy: **34.25%**
Testing accuracy: **33.97%**

Accuracies using K Means is very less. This means that training data is not being clustered into different clusters, which is because this dataset might not have the points for each label clustered around each other and might be scattered around in space, not forming clusters using closest distance from centroid.

(iii)

max_iter ->	**10**	**20**	**30**	**40**	**50**
Training Accuracy (%)	33.79	36.72	35.87	34.84	34.91
Test Accuracy (%)	33.76	36.52	35.65	34.56	34.62



b)

Training accuracy: **96.74%**
Testing accuracy: **82.67%**

Using internal cross-validation, I got the optimal value of **C=3**

I also scaled the data between 0 and 1. Without scaling, the libsvm optimization was taking a lot of time and even the accuracy I was getting was less.

c)

Training accuracy: **99.99%**
Testing accuracy: **78.06%**

Optimal number of hidden units are 1024.

On increasing the number of hidden units the overfitting of training data increases, and taking the number of hidden units too small also doesn't help and gives very less accuracy which is the case of underfitting. This is because we have a lot of features in our dataset, so would need greater number of hidden layer units.
On increasing the batch size, the accuracy of test data decreased but the running time became faster.

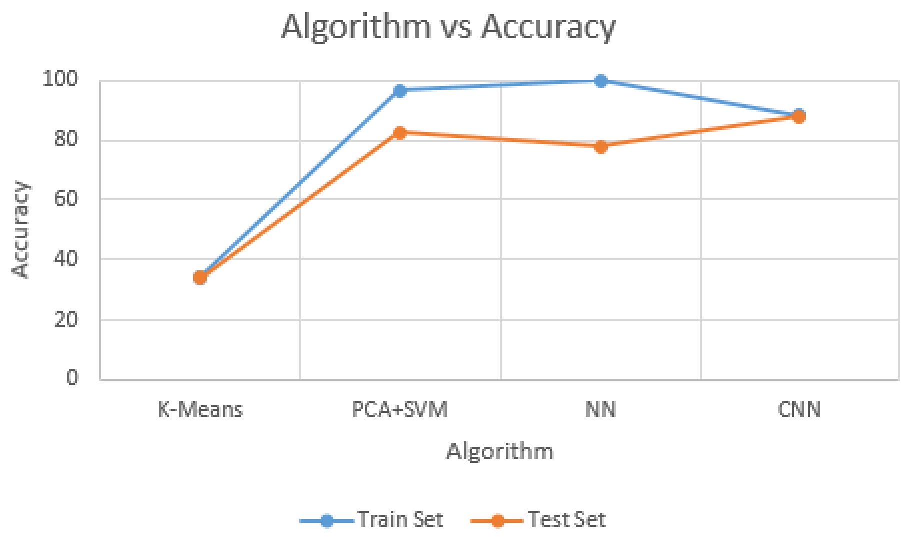
d)

Training accuracy: **95.36%**
Testing accuracy: **89.88%**

Best set of parameters are:
Number of kernels: 32
Kernel size: 5x5
Pool size: 3x3
Strides: 2x2
Number of units in fully connected layer: 1024

e)

	K-Means	**PCA+SVM**	**NN**	**CNN**
Training Accuracy (%)	34.25	96.74	99.99	95.36
Test Accuracy (%)	33.97	82.67	78.06	89.88



We can observe that the worst training and test accuracy is given by K Means, and best test set accuracy is given by deep leaning model of CNN. In neural network, there is overfitting of the training data so the test accuracy is slightly less than SVM and CNN.
CNN outperforms all the models because of its deep learning properties, it can learn from any orientation or size of the images. It learns patterns across space and how these patterns can be aggregated to form objects in images.

2.

https://owncloud.iitd.ac.in/owncloud/index.php/s/EFEqbDanCB7Dn83

Libraries used: sklearn and keras

My architecture consists of a two hidden layers with number of filters as 32, filter size as 5x5, activation function 'relu' followed by a Max Pooling layer with size 3x3. And finally there is a dropout of 25%. Then there is a fully connected layer having 1024 hidden units. And then the final layer with 'softmax' as the activation function and having 20 units.

For training, saving the model and testing, I have a function named 'Q4' which takes training data, training labels, testing data, validation data, validation labels, name of the model file to be loaded, name of the output file for predictions respectively as arguments.

For testing using pre-trained keras model, we can use the function 'predict_from_keras_model' which takes name of the model file to be loaded, test data, name of the output file for predictions as the arguments.