Created By:
Koustav Das (2015CSB1017)
Sachin Bijalwan (2015CSB1027)

Report

Part 1

We have used the inbuilt matlab Kmeans clustering algorithm. It is used as follows
idx=kmeans(X,c)
where X is the dataset and c is the number of clusters.
We have clustered the 5000 image dataset on the different cluster and these are the results generated

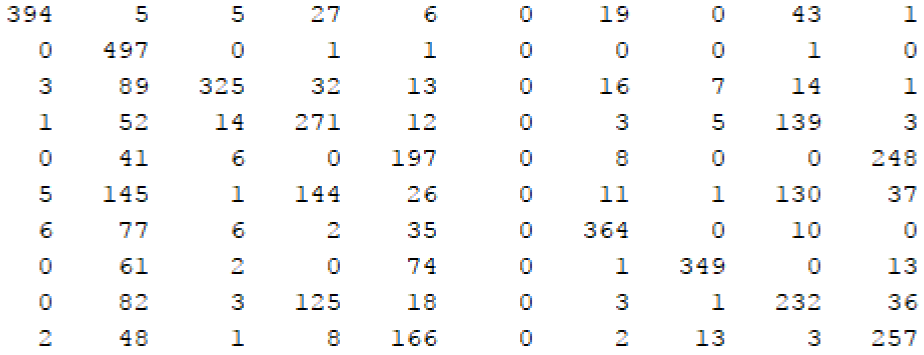
For 10 Cluster

Accuracy : 57.77%

The confusion matrix

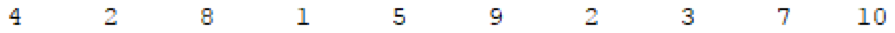
Predicted

Actual



```
   394     5     5    27     6     0    19     0    43     1
     0   497     0     1     1     0     0     0     1     0
     3    89   325    32    13     0    16     7    14     1
     1    52    14   271    12     0     3     5   139     3
     0    41     6     0   197     0     8     0     0   248
     5   145     1   144    26     0    11     1   130    37
     6    77     6     2    35     0   364     0    10     0
     0    61     2     0    74     0     1   349     0    13
     0    82     3   125    18     0     3     1   232    36
     2    48     1     8   166     0     2    13     3   257
```

Class labels that were assigned to each cluster :



```
  4     2     8     1     5     9     2     3     7    10
```

Note: We have used 1 indexing. Hence the highest label that gets assigned to a cluster is 10. Actually this corresponds to 9.

Values that are merging

1. 4 is merging with 9
2. 5 is merging with 1 and 3

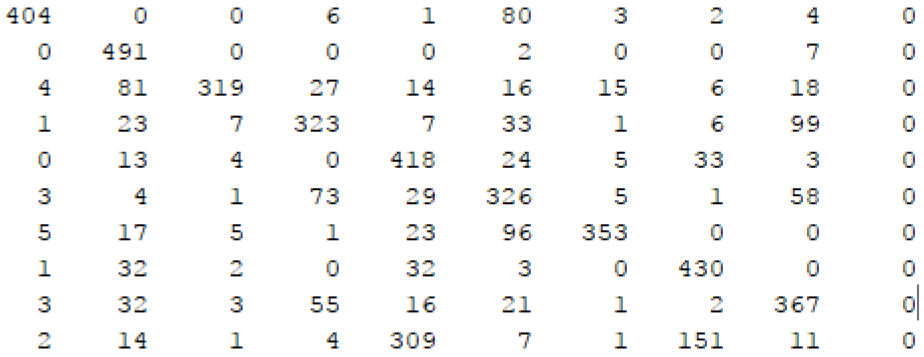
For 15 cluster

Accuray: 68.62%

The confusion matrix

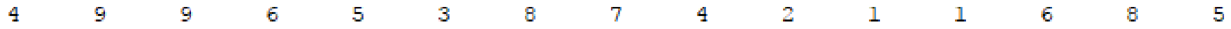
Predicted

Actual



```
404     0     0     6     1    80     3     2     4     0
  0   491     0     0     0     2     0     0     7     0
  4    81   319    27    14    16    15     6    18     0
  1    23     7   323     7    33     1     6    99     0
  0    13     4     0   418    24     5    33     3     0
  3     4     1    73    29   326     5     1    58     0
  5    17     5     1    23    96   353     0     0     0
  1    32     2     0    32     3     0   430     0     0
  3    32     3    55    16    21     1     2   367     0
  2    14     1     4   309     7     1   151    11     0
```

Class labels that were assigned to each cluster :



```
4    9    9    6    5    3    8    7    4    2    1    1    6    8    5
```

Values that are merging

1. 9 is merging with 4. They also look the same.

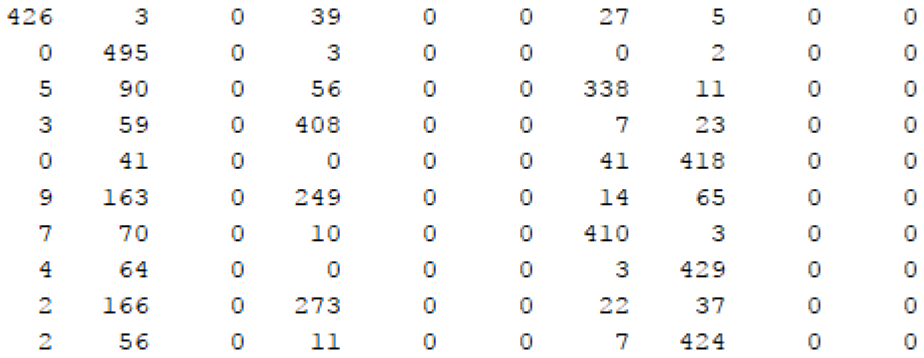
For 5 cluster:

Accuray: 43.36%

The confusion matrix

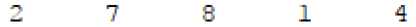
Predicted

Actual



```
 426     3     0    39     0     0    27     5     0     0
   0   495     0     3     0     0     0     2     0     0
   5    90     0    56     0     0   338    11     0     0
   3    59     0   408     0     0     7    23     0     0
   0    41     0     0     0     0    41   418     0     0
   9   163     0   249     0     0    14    65     0     0
   7    70     0    10     0     0   410     3     0     0
   4    64     0     0     0     0     3   429     0     0
   2   166     0   273     0     0    22    37     0     0
   2    56     0    11     0     0     7   424     0     0
```

Class labels that were assigned to each cluster :



```
  2     7     8     1     4
```

Values that are merging

1. 9 is merging up with 7
2. 2 is merging up with 6
3. 4 is merging up with 7
4. 8 is merging up with 1 and 3
5. 5 is merging up with 1 and 3

Part 2

What is principal component analysis?

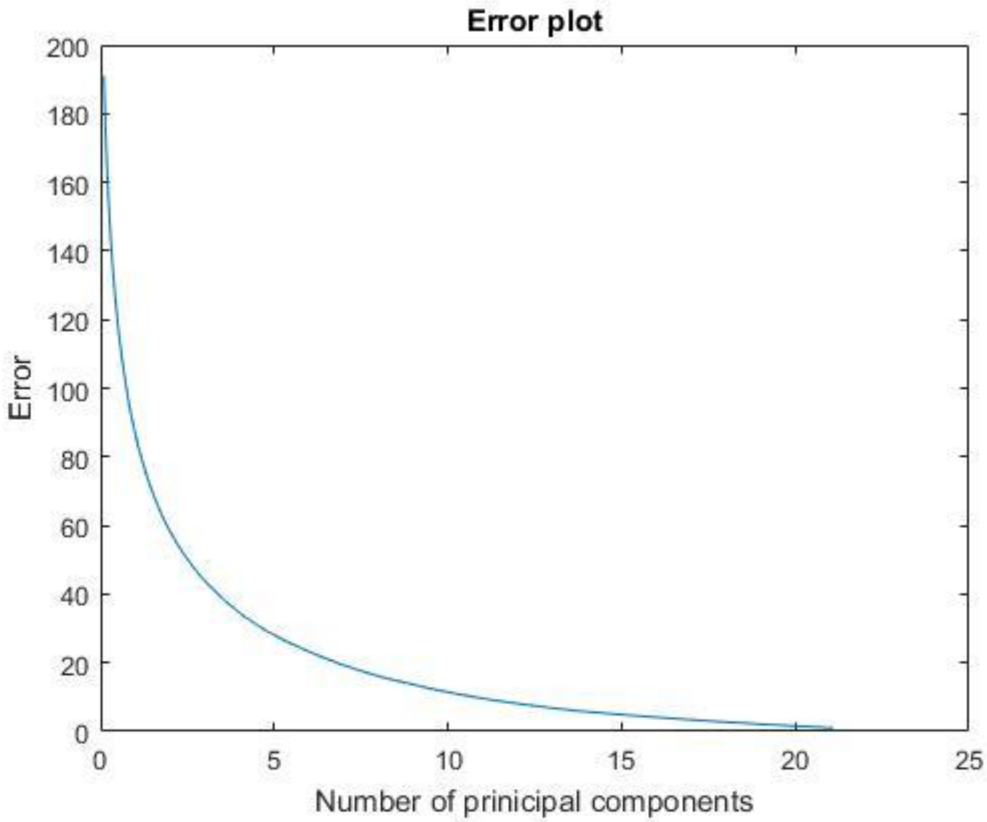
This is a dimensionality reduction technique. Here we project the data onto lower dimension such that these dimension correspond to the maximum variation in the values.

How it is implemented in this assignment?

We found the covariance matrix. Then using this matrix we generate all the 400 dimension of the transformed space(all the principal components). We then reconstruct the image by taking only a few principal components. This creates some reconstruction error.

For reaching the reconstruction error of 0.1 we need **191** principal components.

This is the error plot that has been generated



Observations

1. We find that on increasing the number of principal components the reconstruction error decreases exponentially.
2. Hence the reconstructed image is getting clearer.

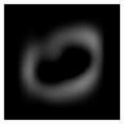
Reconstruction of the image

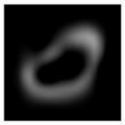
For 0

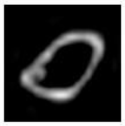
The first reconstructed image is for principal component 2

The first reconstructed image is for principal component 3

The first reconstructed image is for principal component 100





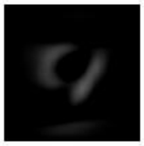


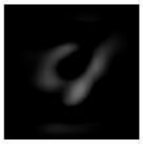
For 7

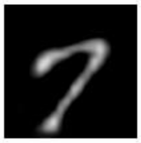
The first reconstructed image is for principal component 2

The first reconstructed image is for principal component 3

The first reconstructed image is for principal component 100





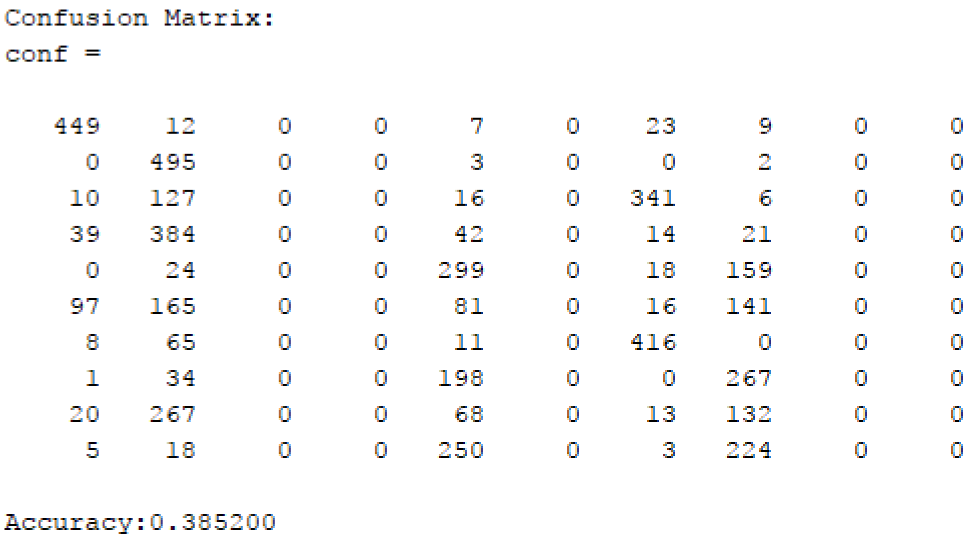


Observations

1. The image is getting clearer as the number of principal components is increased since the amount of reconstruction error is getting decreased.

Part 3

For 5 Clusters

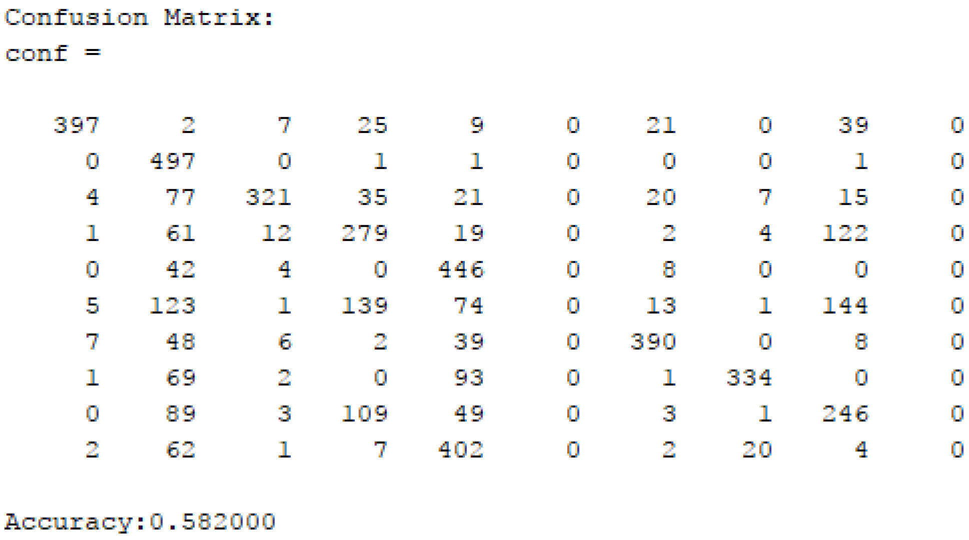


```
Confusion Matrix:
conf =

   449    12     0     0     7     0    23     9     0     0
     0   495     0     0     3     0     0     2     0     0
    10   127     0     0    16     0   341     6     0     0
    39   384     0     0    42     0    14    21     0     0
     0    24     0     0   299     0    18   159     0     0
    97   165     0     0    81     0    16   141     0     0
     8    65     0     0    11     0   416     0     0     0
     1    34     0     0   198     0     0   267     0     0
    20   267     0     0    68     0    13   132     0     0
     5    18     0     0   250     0     3   224     0     0

Accuracy:0.385200
```

For 10 Clusters

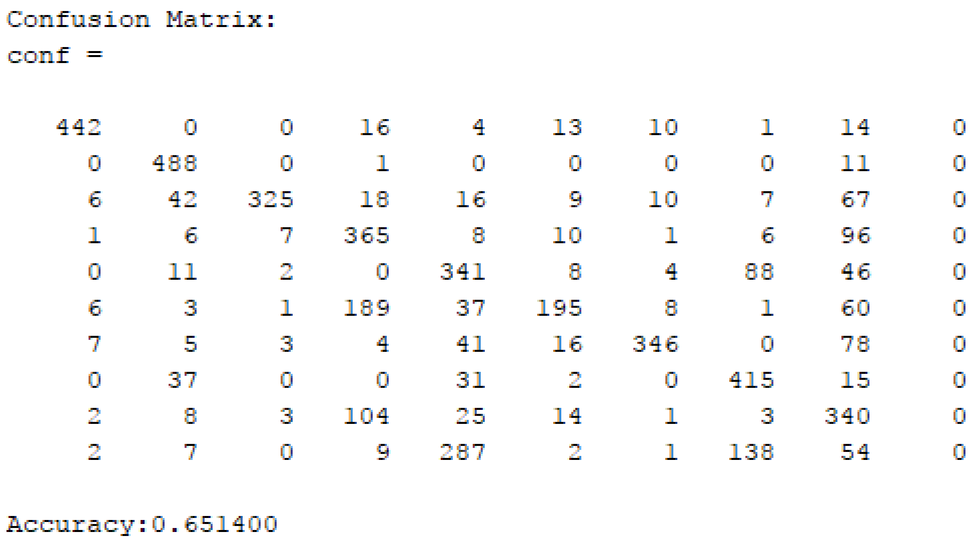


```
Confusion Matrix:
conf =

   397     2     7    25     9     0    21     0    39     0
     0   497     0     1     1     0     0     0     1     0
     4    77   321    35    21     0    20     7    15     0
     1    61    12   279    19     0     2     4   122     0
     0    42     4     0   446     0     8     0     0     0
     5   123     1   139    74     0    13     1   144     0
     7    48     6     2    39     0   390     0     8     0
     1    69     2     0    93     0     1   334     0     0
     0    89     3   109    49     0     3     1   246     0
     2    62     1     7   402     0     2    20     4     0

Accuracy:0.582000
```

For 15 Clusters



```
Confusion Matrix:
conf =

   442     0     0    16     4    13    10     1    14     0
     0   488     0     1     0     0     0     0    11     0
     6    42   325    18    16     9    10     7    67     0
     1     6     7   365     8    10     1     6    96     0
     0    11     2     0   341     8     4    88    46     0
     6     3     1   189    37   195     8     1    60     0
     7     5     3     4    41    16   346     0    78     0
     0    37     0     0    31     2     0   415    15     0
     2     8     3   104    25    14     1     3   340     0
     2     7     0     9   287     2     1   138    54     0

Accuracy:0.651400
```

Observation

1. It can be seen that in some cases the accuracy of Kmeans clustering is actually increasing. This can happen due to the fact that unwanted noise is getting removed because of the reconstruction and only the important information is left on which the clustering algorithm is performing better.