CSCI 5521 - INTRO TO MACHINE LEARNING

TANVEER SINGH VIRDI

ASSIGNMENT 2

OCTOBER 15, 2019

PROBLEM 1:

a.

INTRO TO MACHINE LEARNING

ASSIGNMENT 2

QUESTION 1.

SOLUTION. Given,

Multivariate Gaussian Distribution:

$$p(x|C_i) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)\right)$$

The discriminant function is written as -

$$g_i(x) = \log p(x|C_i) + \log P(C_i)$$

$$= -\frac{D}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_i| - \frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i) + \log P(C_i)$$

a.) Model 2:

$S = S1 = S2$, i.e, the covariance is shared between the two classes.

For class 1, the log likelihood becomes -

$$L(\mu_1, S|x) = \log l(\mu_1, S|x)$$

$$= \sum_{t=1}^{N_1} \log P(x^t | \mu_1, S)$$

$$= \sum_{t=1}^{N_1} \log\left(\frac{1}{(2\pi)^{D/2}|S|^{1/2}} \exp\left(-\frac{1}{2}(x^t-\mu_1)^T S^{-1}(x^t-\mu_1)\right)\right)$$

$$= \sum_{t=1}^{N_1}\left[-\frac{D}{2}\log 2\pi - \frac{1}{2}\log|S|\right] + \sum_{t=1}^{N_1}\left(-\frac{1}{2}(x^t-\mu_1)^T S^{-1}(x^t-\mu_1)\right)$$

$$= -\frac{N_1 D}{2}\log 2\pi - \frac{N_1}{2}\log|S| - \frac{1}{2}\sum_{t=1}^{N_1}(x^t-\mu_1)^T S^{-1}(x^t-\mu_1)$$

Differentiating with respect to S^{-1} we get -

$$\frac{d}{dS^{-1}}\left(-\frac{N_1 D}{2}\log 2\pi - \frac{N_1}{2}\log|S|\right) - \frac{1}{2}\frac{d}{dS^{-1}}\sum_{t=1}^{N_1}(x^t-\mu_1)^T S^{-1}(x^t-\mu_1)$$

$$= -\frac{N_1}{2}\frac{d}{dS^{-1}}\log|S| - \frac{1}{2}\sum_{t=1}^{N_1}(x^t-\mu_1)^T(x^t-\mu_1)$$

$$= \frac{N_1}{2} S - \frac{1}{2} \sum_{t=1}^{N_1} (x^t - \mu_1)^T (x^t - \mu_1) \quad -① \quad \left[\text{Using } \frac{d}{dx} |X^{-1}| = -|X^{-1}| (X^{-1})^T \right.$$

Similarly if we compute the log likelihood for class 2 and differentiate the result with respect to S^{-1}, we get -

$$\frac{N_2}{2} S - \frac{1}{2} \sum_{t=1}^{N_2} (x^t - \mu_2)^T (x^t - \mu_2) \quad -②$$

Combining ① and ② and setting the result to 0, we get -

$$\frac{N_1}{2} S - \frac{1}{2} \sum_{t=1}^{N_1} (x^t - \mu_1)^T (x^t - \mu_1) + \frac{N_2}{2} S - \frac{1}{2} \sum_{t=1}^{N_2} (x^t - \mu_2)^T (x^t - \mu_2) = 0$$

$$\Rightarrow \left(\frac{N_1 + N_2}{2} \right) S = \frac{1}{2} \left[\sum_{t=1}^{N_1} (x^t - \mu_1)^T (x^t - \mu_1) + \sum_{t=1}^{N_2} (x^t - \mu_2)^T (x^t - \mu_2) \right]$$

$$\Rightarrow S = \frac{1}{N_1 + N_2} \left[\sum_{t=1}^{N_1} (x^t - \mu_1)^T (x^t - \mu_1) + \sum_{t=1}^{N_2} (x^t - \mu_2)^T (x^t - \mu_2) \right]$$

Where N_1 and N_2 are the number of training instances in class 1 and class 2 respectively. We can rewrite the above equation as:

$$S = \frac{N_1}{N_1 + N_2} \sum_{t=1}^{N_1} \frac{(x^t - \mu_1)^T (x^t - \mu_1)}{N_1} + \frac{N_2}{N_1 + N_2} \sum_{t=1}^{N_2} \frac{(x^t - \mu_2)(x^t - \mu_2)}{N_2}$$

We know,

the prior probability for class 1, $P(C_1) = \frac{N_1}{N_1 + N_2}$

the prior probability for class 2, $P(C_2) = \frac{N_2}{N_1 + N_2}$

Covariance for class 1, $S_1 = \sum_{t=1}^{N_1} (x^t - \mu_1)^T (x^t - \mu_1) / N_1$

Covariance for class 2, $S_2 = \sum_{t=1}^{N_2} (x^t - \mu_2)^T (x^t - \mu_2) / N_2$

Then the equation for S becomes -

$$\boxed{S = P(C_1) S_1 + P(C_2) S_2}$$

Model 3 :

S_1 and S_2 are diagonal matrices.

$$S_1 = \alpha_1 I$$

$$S_2 = \alpha_2 I$$

For class 1, the log likelihood becomes -

$$\mathcal{L}(M_1, \alpha_1 | x) = \log l(M_1, \alpha_1 | x)$$

$$= \sum_{t=1}^{N_1} \log p(x | M_1, \alpha_1)$$

$$= \sum_{t=1}^{N_1} \log \left(\frac{1}{(2\pi)^{D/2} |\alpha_1 I|^{1/2}} \exp\left(-\frac{1}{2} (x^t - M_1)^T (\alpha_1 I)^{-1} (x^t - M_1) \right) \right)$$

$$= \sum_{t=1}^{N_1} \log \left(\frac{1}{(2\pi)^{D/2} (\alpha_1)^{1/2}} \exp\left(-\frac{1}{2} \frac{(x^t - M_1)^T (x^t - M_1)}{\alpha_1} \right) \right)$$

$$= \sum_{t=1}^{N_1} \log \left(\frac{1}{(2\pi)^{D/2} (\alpha_1)^{1/2}} \exp\left(-\frac{1}{2} \sum_{j=1}^{D} \frac{(x_j^t - M_{1j})^2}{\alpha_1} \right) \right)$$

$$= \sum_{t=1}^{N_1} \left[-\frac{D}{2} \log 2\pi - \frac{1}{2} \log \alpha_1 \right] - \frac{1}{2} \sum_{t=1}^{N_1} \sum_{j=1}^{D} \frac{(x_j^t - M_{1j})^2}{\alpha_1}$$

$$= -\frac{N_1 D}{2} \log 2\pi - \frac{N_1}{2} \log \alpha_1 - \frac{1}{2} \sum_{t=1}^{N_1} \sum_{j=1}^{D} \frac{(x_j^t - M_{1j})^2}{\alpha_1}$$

Differentiating with respect to α_1, we get -

$$\frac{d}{d\alpha_1} \left[-\frac{N_1 D}{2} \log 2\pi - \frac{N_1}{2} \log \alpha_1 \right] - \frac{1}{2} \frac{d}{d\alpha_1} \sum_{t=1}^{N_1} \sum_{j=1}^{D} \frac{(x_j^t - M_{1j})^2}{\alpha_1}$$

$$= -\frac{N_1}{2\alpha_1} + \frac{1}{2} \sum_{t=1}^{N_1} \sum_{j=1}^{D} \frac{(x_j^t - M_{1j})^2}{\alpha_1^2} \quad \text{— (1)}$$

Similarly if we compute the log likelihood for class 2 and differentiate the result with respect to α_2, we get -

$$-\frac{N_2}{2\alpha_2} + \frac{1}{2} \sum_{t=1}^{N_2} \sum_{j=1}^{D} \frac{(x_j^t - M_{2j})^2}{\alpha_2^2} \quad \text{— (2)}$$

N_1 and N_2 are the number of training instances in class 1 and class 2 respectively.

Solving for α_1 and α_2 by setting the results in ① and ② to 0, we get -

$$\alpha_1 = \frac{1}{D} \sum_{j=1}^{D} \sum_{t=1}^{N_1} \frac{(x_j^t - M_{1j})^2}{N_1}$$

$$\alpha_2 = \frac{1}{D} \sum_{j=1}^{D} \sum_{t=1}^{N_2} \frac{(x_j^t - M_{2j})^2}{N_2}$$

Thus,

$$S_1 = \alpha_1 I = \frac{1}{D} \sum_{j=1}^{D} \sum_{t=1}^{N_1} \frac{(x_j^t - M_{1j})^2}{N_1} I$$

$$S_2 = \alpha_2 I = \frac{1}{D} \sum_{j=1}^{D} \sum_{t=1}^{N_2} \frac{(x_j^t - M_{2j})^2}{N_2} I$$

b. Programming Question

c. Error rates for Multivariate Gaussian are as follows:

	Model 1	Model 2	Model 3
Test Set 1	0.22	0.17	0.26
Test Set 2	0.23	0.55	0.61
Test Set 3	0.11	0.45	0.27

We have used 3 different models to classify training instances in 3 different test sets. We see that model 1 has the least error rate among the 3 different models followed by model 2 and model 3. Model 1 is the most complex among the three models because it has a large number of parameters to estimate. In general, For K different classes and d dimensional data, it has to calculate K.d means and K.d(d+1)/2 variances. Because of this it has a quadratic discriminant function and is able to explain the complexity of the problem represented by the data at hand much better than the other two models. In model 2 we assume that the covariance matrix is shared among the classes. In midel 2 we just need to calculate d(d+1)/2 variances and K.d means, which is smaller than the number of parameters associated with Model 1. Because of this Model 2 has a linear discriminant function and is less effective in explaining the complexity of the data than Model 1 and hence has higher error rates. For model 3 we have a diagonal covariance matrix for each class. The features are independent of each other and model 3 just has to calculate the K.d means and the K alpha values. Because of this model 3 is least effective in explaining the complexity in the data and runs the risk of introducing bias. Hence error rates for model 3 are the worst among the 3 models.

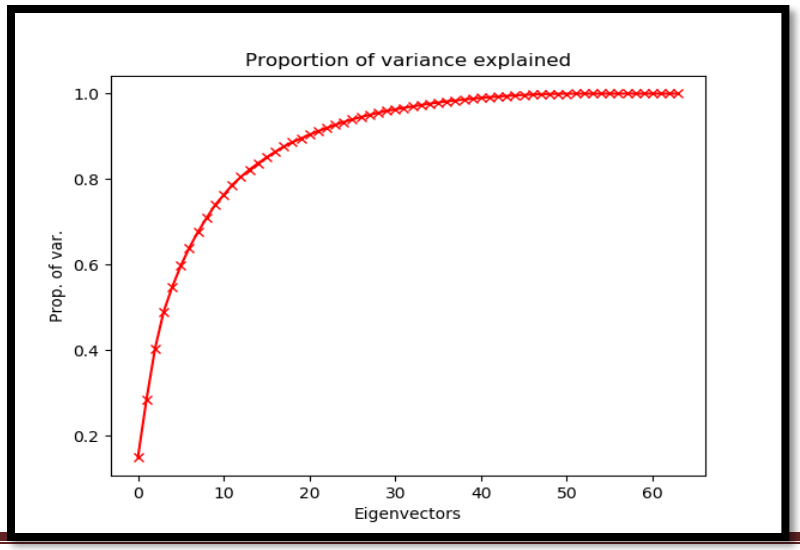
PROBLEM 2:

SOLUTION:

a. Error rate for KNN are as follows :

K=1	K=3	K=5	K=7
0.05387	0.0404	0.0437	0.05387

b. Plot of proportion of variance :

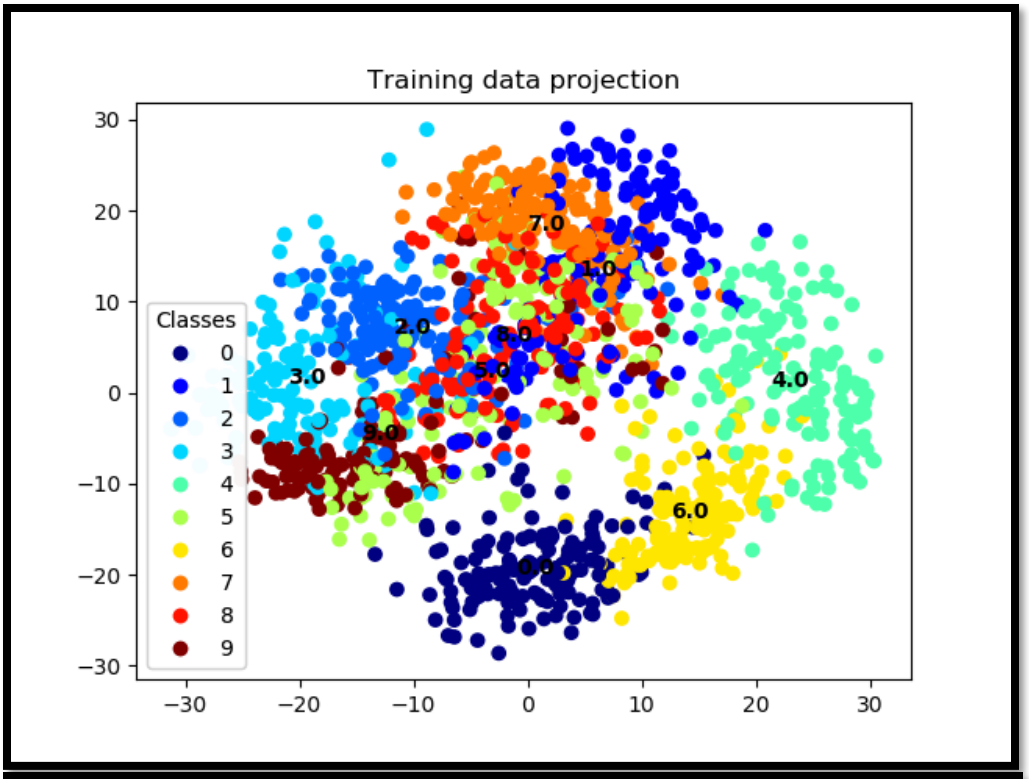


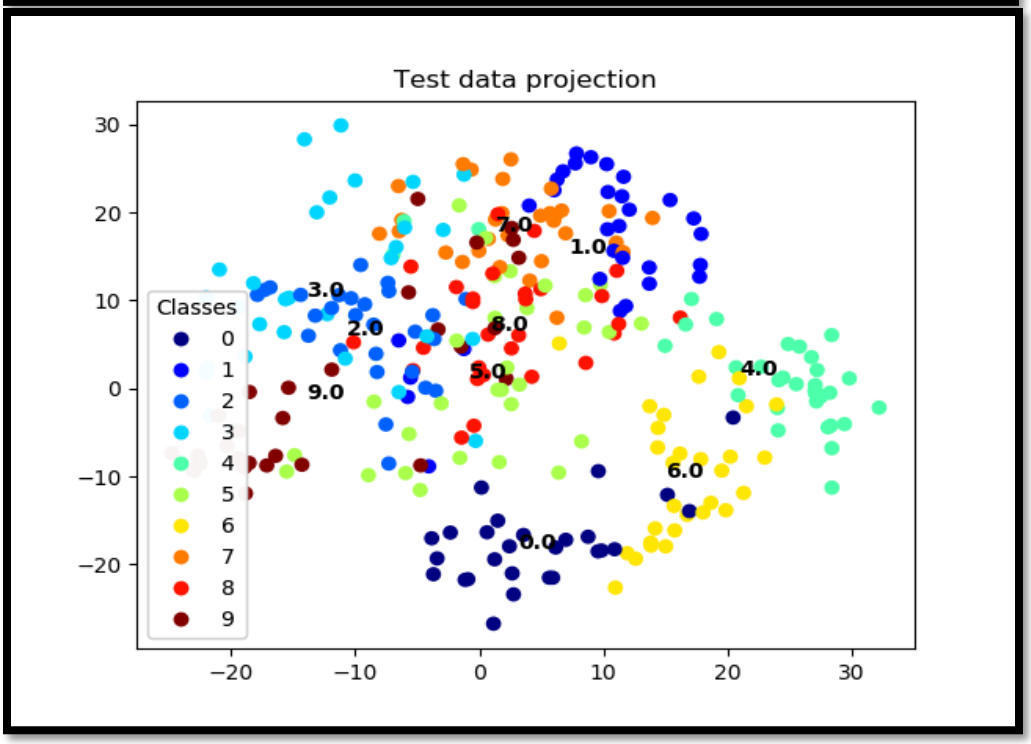
Minimum number of eigenvectors that explain 90% of the variance = 21

KNN error rate after projecting the data onto K=21 principal components :

K=1	K=3	K=5	K=7
0.04713	0.04713	0.05387	0.05387

c. Projection of test data and training data on 2 dimensions using PCA :

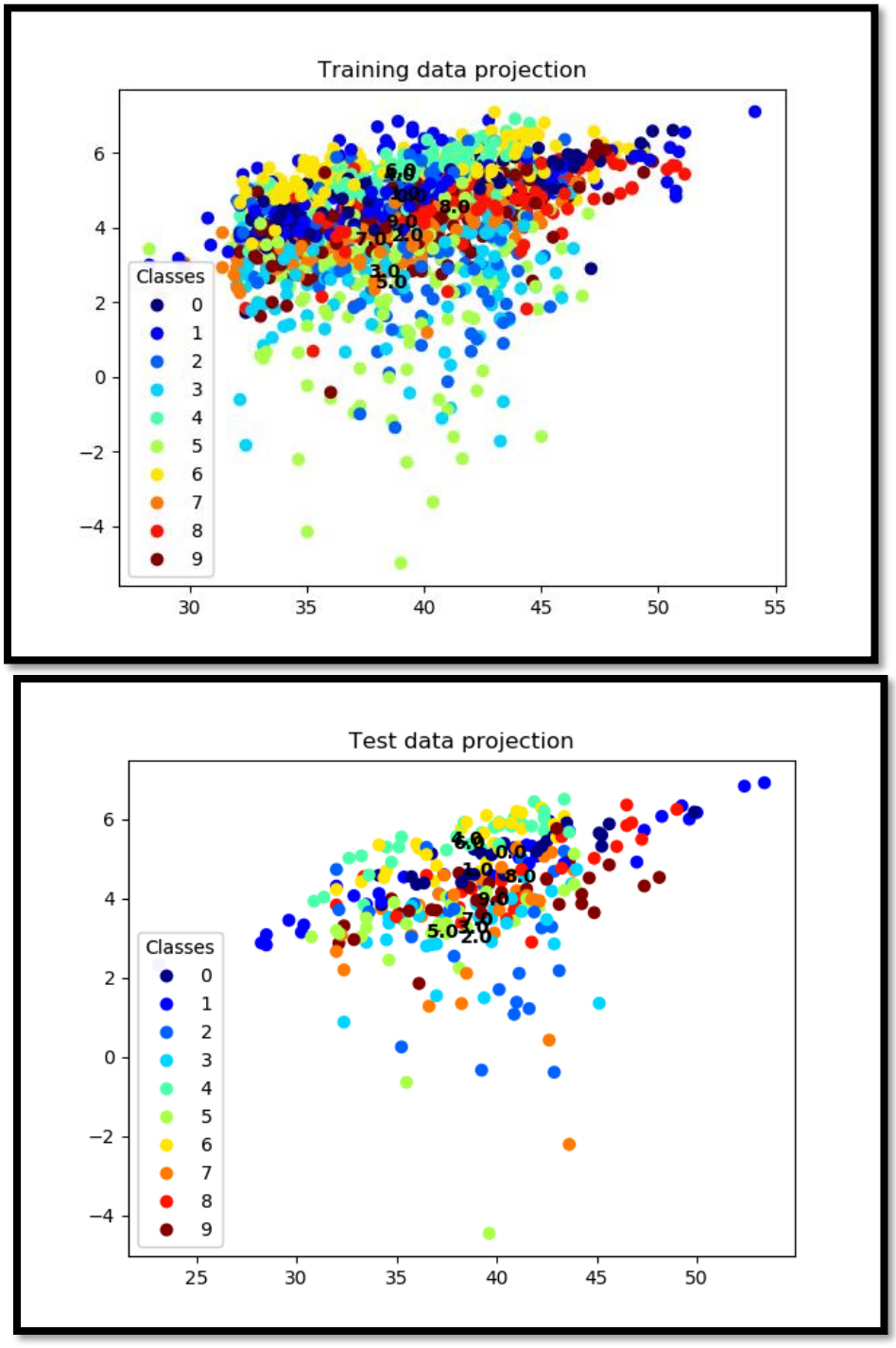




d. KNN error rate after projecting the training data into L=[2,4,9] dimensions :

	K = 1	K = 3	K = 5
L = 2	0.7508	0.7575	0.7340
L = 4	0.6127	0.5925	0.5589
L = 9	0.2390	0.2424	0.2356

e. Projection of training and test data into 2 dimensions using LDA :

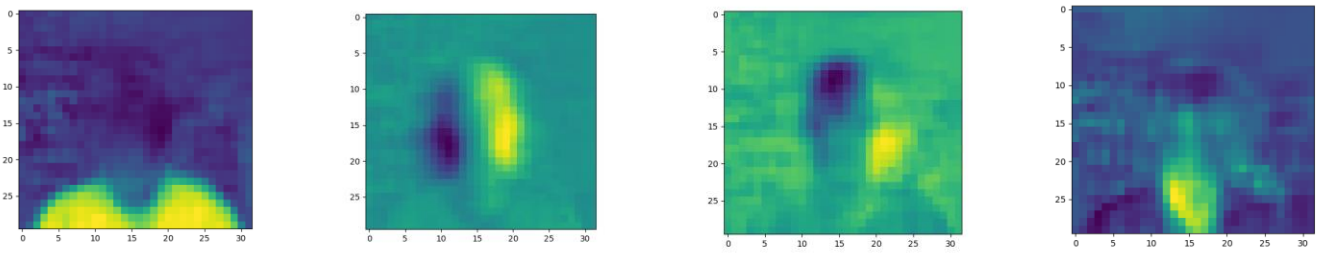


PROBLEM 3:

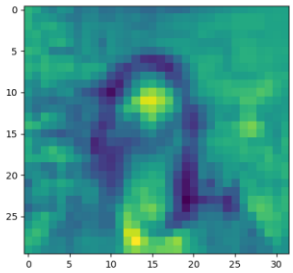
SOLUTION:

a. The first five eigenfaces obtained using PCA are :

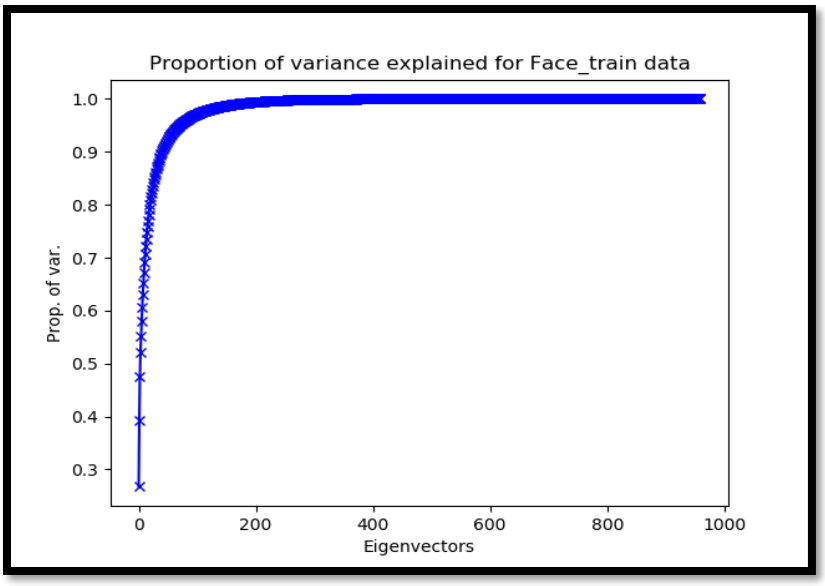
1 2 3 4



5



b. Plot of proportion of variance :



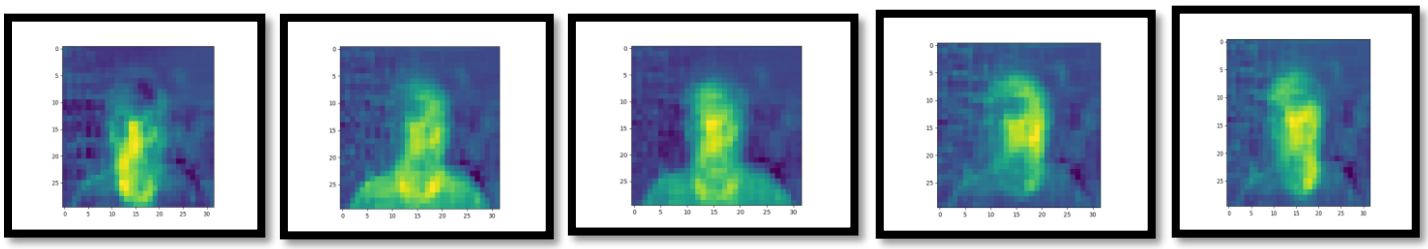
Minimum number of eigenvectors that explain 90% of the variance = 41

KNN error rate after projecting the data onto K=41 principal components :

K=1	K=3	K=5	K=7
0.1129	0.2338	0.4112	0.4354

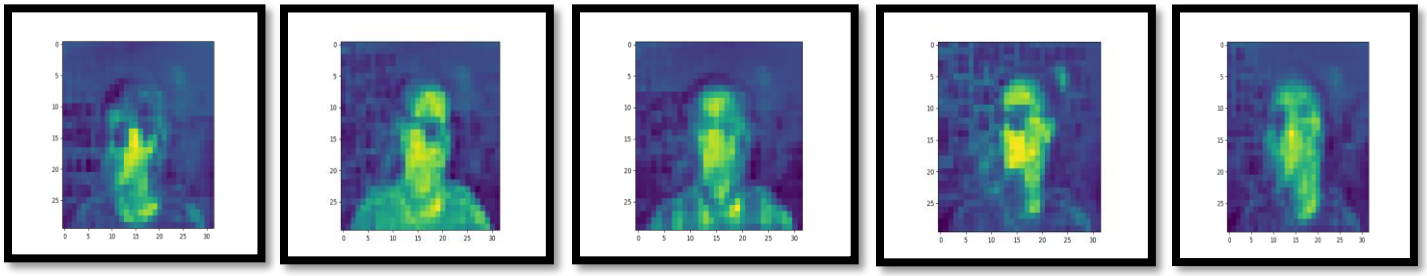
c. For K = 10, reconstructed image :

1 2 3 4 5



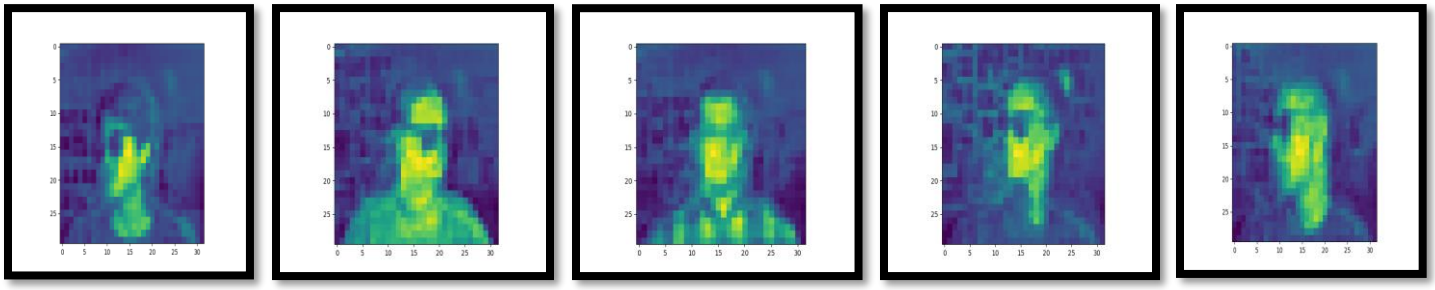
For K = 50, reconstructed image :

1 2 3 4 5



For K = 100 , reconstructed image :

1 2 3 4 5



From the above results, we see that when the images were reconstructed using 10 principal components, the images are not very sharp and clear. The image does not show very clearly whether the subject is wearing a sunglass or not. When the images were reconstructed using 50 principal components and since 90% proportion of variance is explained by 41 principal components, we see that the images look much clearer. We can roughly make out whether the subject is wearing sunglass or not. Finally when the images were reconstructed using 100 principal components, it gave the best results and the images looked really sharp and clear. All the pictures show clearly whether the subject is wearing sunglass or not.