PROJECT 1

Pavithra Ezhilarasan
UID: 114842763

AIM: To implement different classifiers to achieve face recognition

PROCESS:

Training and test data:

All experiments were carried out on Data.mat dataset file. The training data comprises of the neutral face and the face with expressions (first two images of each subject). The test data comprises of the face under a different illumination (third image of the subject).

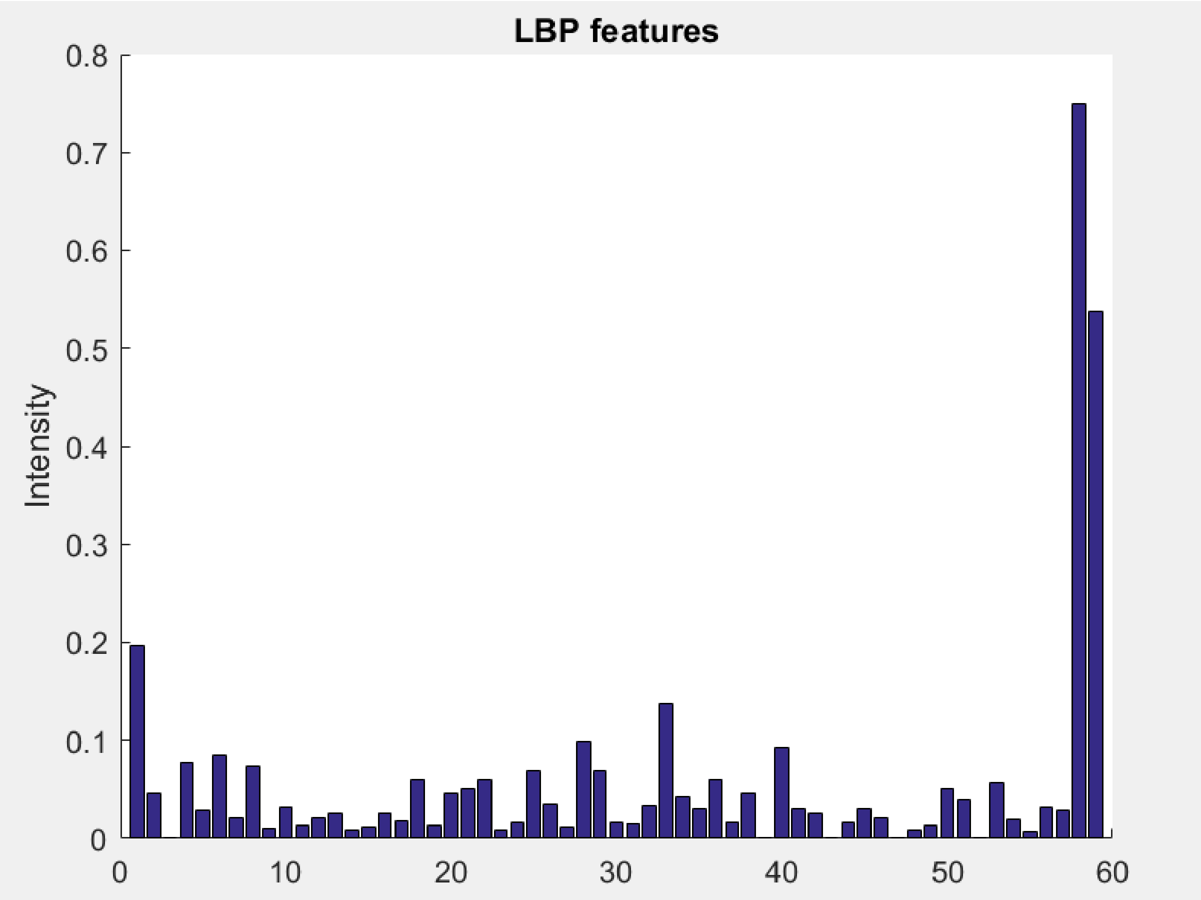
features.m

This code sets up the database on which all other classifiers are to run. The images are pre-processed in order to be able to discern features well. The images are subjected to an adaptive histogram equalizer, followed by image sharpening and adjusting. Adaptive histogram equalizer enhances the contrast of the grayscale image by transforming the values using contrast-limited adaptive histogram equalization (CLAHE). The imsharpen() function returns an enhanced version of the grayscale input image, where the image features, such as edges, have been sharpened using the unsharp masking method. The imadjust() function maps the intensity values in grayscale image to new values such that 1% of data is saturated at low and high intensities of the original image. This increases the contrast of the output image. Below is a figure of the first face image (neutral face of the first subject) before and after pre-processing.

Adaptive histogram equivalent is used instead of histogram equivalent because it performs contrast-limited adaptive histogram equalization. Unlike histogram equivalent, it operates on small data regions (tiles) rather than the entire image. Each tile's contrast is enhanced so that the histogram of each output region approximately matches the specified histogram (uniform distribution by default). The contrast enhancement can be limited in order to avoid amplifying the noise which might be present in the image.

The dataset is split into X_train and X_test. The Y (class) labels are also accordingly generated and stored. X_train comprises 2/3 of the entire dataset and X_test comprises of the remaining 1/3 of the dataset.

Linear Binary Pattern (LBP) features are extracted from the training and test data and stored separately. The LBP feature extraction method was chosen because it is fairly invariant to pose and illumination. The LBP features were extracted into 59 different bins. The following diagram is an LBP representation of the LBP features obtained for the normal face of the first subject i.e face(:,:,1)

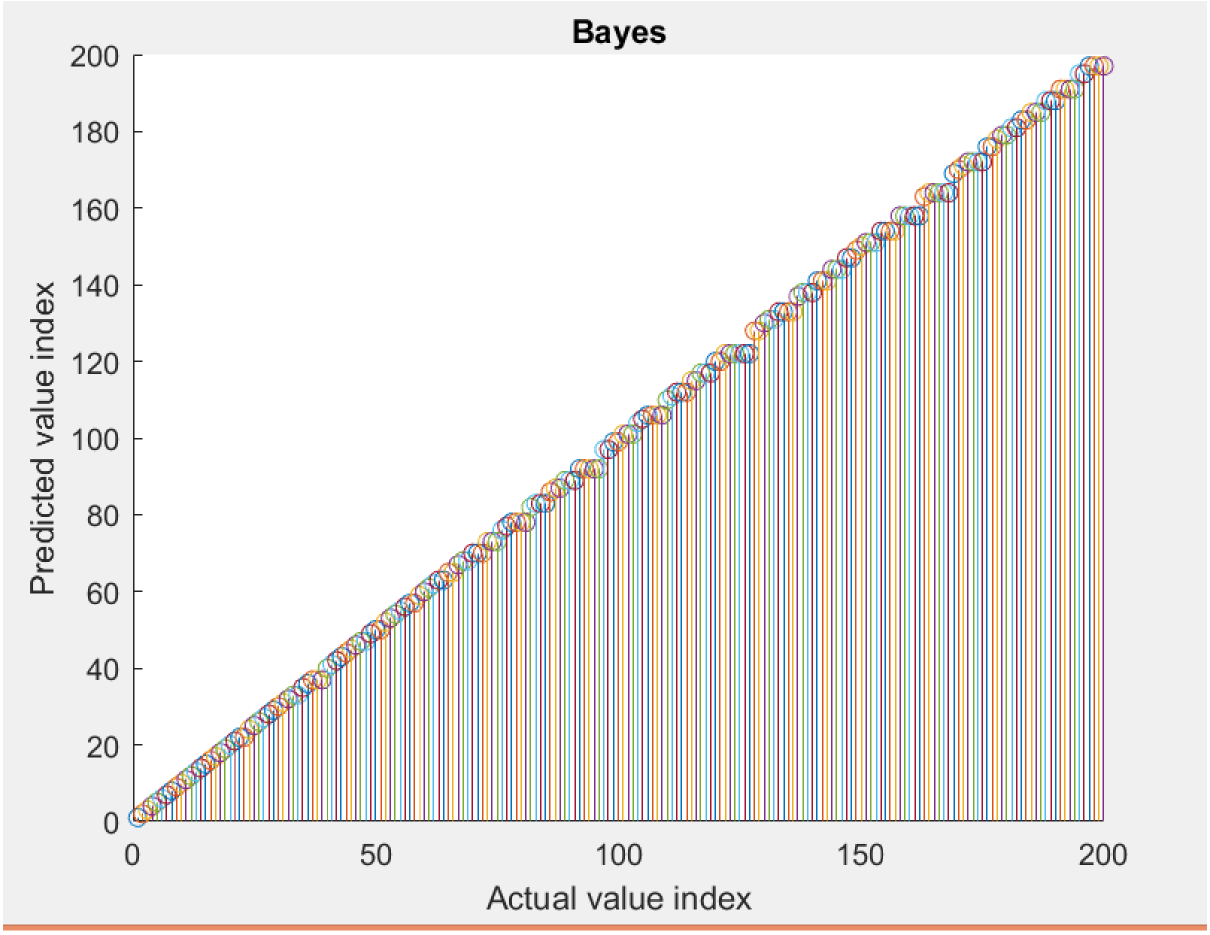


Across the entire dataset, it is noticed that the intensity for the latter bins are more and hence we give more importance to these features by weighting them.

The LBP feature values are small and to avoid miscalculation while dealing with small floating point values, the values are converted to logarithmic scale. It was also noticed that some bins had zero contribution to the image. These bins will pose a problem while getting converted to the logarithmic scale. Hence, we add the mean value of all the features of the image to the bins. Now, the features are stored into variable lbp_train and lbp_test for later use. The variables are saved so that they can be accessed later and save on computing time.

BAYES CLASSIFIER

In the Bayes classifier, the features database is loaded. The mean and covariance of the training data is computed. The Bayes classifier is implemented assuming a Gaussian distribution and calculating the class likelihood with the maximum probability. Below is a plot of predicted value index against the actual value index. An ideal, 100% accuracy result should give a completely linear plot. Here, an accuracy of 56.5% is observed and hence we see misclassifications which breaks the expected linear trend.



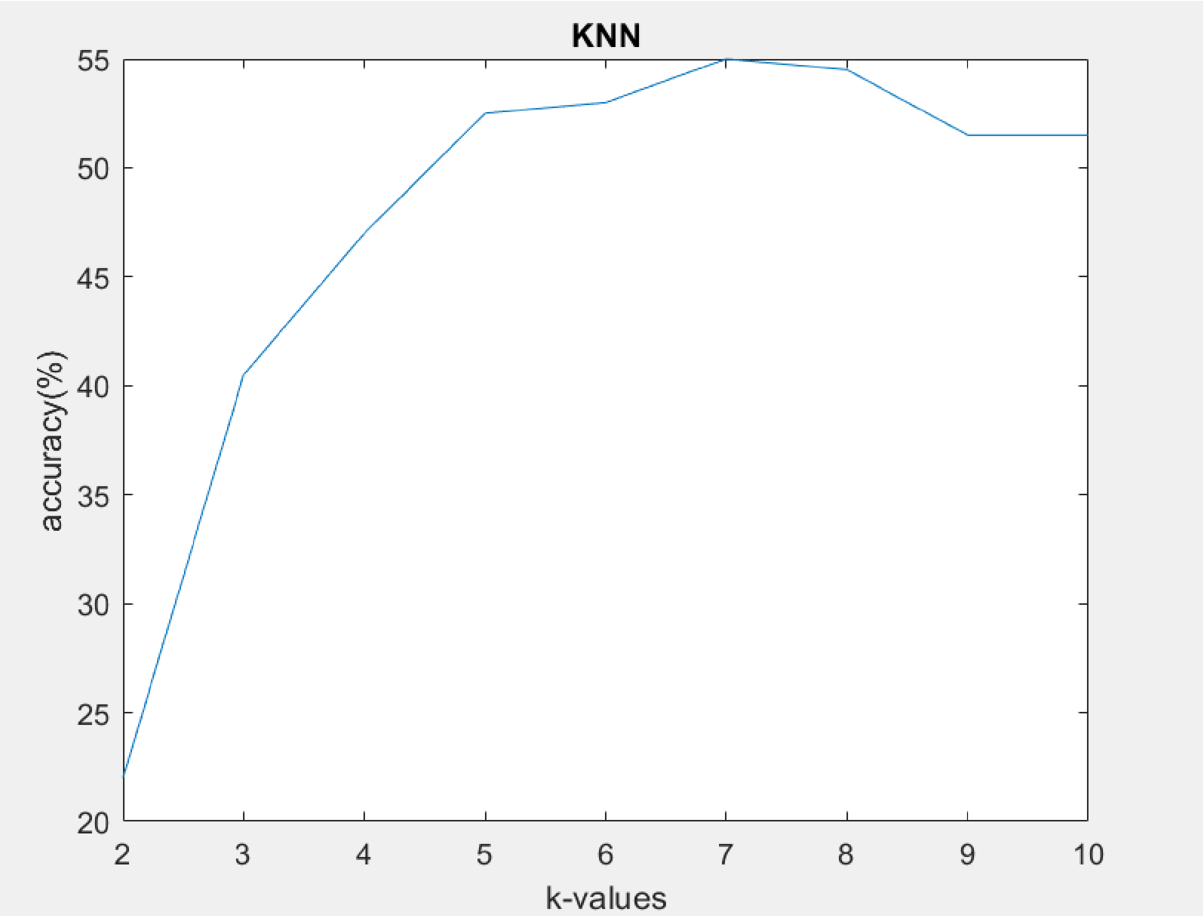
K-NEAREST NEIGHBOUR

In this method, the k-nearest neighbours are chosen according to the distance between the training and test data. In this method, Euclidean distance is chosen as the distance metric. Experimentation was carried out using Mahalanobis distance as a metric. The Mahalanobis distance takes the co-variances into account, which lead to elliptic decision boundaries in the 2D case, as opposed to the circular boundary in the Euclidean case. The Euclidean distance may be seen as a special case of the Mahalanobis distance with equal variances of the variables and zero covariances. Mahalanobis distance as a metric it did not make much difference in the data set used as the two quantities whose distance is to be measured are of the same type. Hence, Euclidean distance was chosen as the distance metric.

It was interesting to see that the accuracy of prediction decreased as the k-value increased. This is in-line with the theory where larger the cluster size, there are more options to choose from and hence there is more room for misclassification to occur.

A vote is taken among the different classes and the one with the majority vote is set as the class for the test sample. In the case of a tie, the one with the shortest distance is chosen. In this case, as we sort the values, in case of a tie, the first value s picked up.

Below, is a pictorial representation of the fall of accuracy of predicting new test samples(%) as k increases. The maximum accuracy is found to be 55% at k=7.



We see that initially the accuracy increases until k=7 and then the accuracy starts decreasing.

Defining k is one the major issues that prevent the use of this simple algorithm more often. The k in k-Nearest Neighbour rule depends on the kind of dataset used. There is no rule to know which range of k should be used. To know the range of k to start with, there is a common rule to start at k=squareroot(n); where n is the number of features.

Interestingly this aligns with the practical results where maximum accuracy is for k=7 which is close to squareroot(59), as the number of features after Linear Binary patter extraction is 59.

PRINCIPAL COMPONENT ANALYSIS:

It is a technique for dimensionality reduction. The dimensions were reduced to 21 to 19 dimensions. The technique of Principal component Analysis (PCA) is primarily used to reduce the computation time while not compromising much on the accuracy. When Bayes classifier was implemented on the dataset, the accuracy increased to 91% compared to Bayes classifier applied on raw data (55%). This is an interesting result because in addition to reduction in computation time the accuracy shoots up too. In theory the PCA does not comment on accuracy improved in addition to reduced computation time, but in practice it improves rate of training, simplifies the required structure to represent the data, and results in systems that better characterize the "intermediate structure" of the data instead of having to account for multiple scales.

An insight I gained from Jonathon Shlens' "A Tutorial on Principal Component Analysis": Performing PCA is like choosing a camera angle, to gain the best possible view of the variance to be explained.

On applying K-NN algorithm after PCA, the accuracy is at 47%. This is a reduction from the original results where all the eigen values is preserved. But this is the cost to be paid to reduced time complexity.

LINEAR DISCRIMINANT ANALYSIS:

This is also a technique of dimensionality reduction. This techniques is different from PCA in the fact that Linear Discriminant Analysis (LDA) is used to achieve separability of the classes. LDA is be used for unsupervised learning. The mean and covariance is therefore calculated across the entire training data. As the covariance can be calculated only for a square matrix, an identity matrix of the relevant size is added to it to get it in the form of a square matrix.

On applying the technique of LDA too, it is observed that not only the computation time but also the accuracy of the classifier improves because of similar reasons mentioned in the above PCA section.

The dimensions used are reduced from 21 to 19 dimensions. On applying a Bayes classifier after LDA, there is an increase in accuracy to 79.5% compared to direct application of Bayes on the raw data. This increase results due to a similar reason mentioned in the above section (PCA).

On applying K-NN algorithm after LDA, the accuracy of prediction for the test data is 52.5%. This is an increase compared to the 47% on applying the k-NN classifier directly on the test data without subjecting it to LDA.

MAIN LEARNING POINTERS ON DOING THIS PROJECT:

- Analysis on different image pre-processing tools and which one is more suitable to the data in hand
- The different scenarios where Mahalanobis distance helps when used as a distance metric compared to Euclidean distance
- Dimensionality reduction techniques like PCA, LDA not only help in computation time reduction but also improve the accuracy depending on the dataset
- Computation of inverse of a non-square matric can be done by adding an identity matrix of an equivalent dimension

CONCLUSION:

Implemented Bayes and K-Nearest Neighbour algorithm for face recognition. Used dimensionality reduction techniques like Principal component analysis and Linear discriminant analysis and observed their effect on the above classifiers.