Answer to the Question number 1

Eigen Faces:

Eigenfaces are considered to be a PCA based recognition. This finds features from image datasets where larger variance is assured. It represents an image as a linear combination of features. Eigenfaces can be calculated using the PCA and also implementing SVD. Here the rows are used to present the features and the instances are presented in the columns. The following steps are used for eigenfaces:

1. Normalizing given dataset
2. Covariance matrix is calculated
3. Eigen vectors are calculated
4. Eigen values are calculated from the covariance matrix
5. Largest eigen value is used in order to get maximum variance.
6. Weight for each image in the dataset is calculated. This is done by projecting data in the new reference frame. This is also considered to be eigenspace. Multiplying the data set matrix by such a matrix whose columns are the eigenvectors referred to the largest eigenvalues.
7. The projected data is used as new variables for future analysis.

Answer to the Question number 2

EigenFaces Implementation

1. Olivetti dataset is used in the assignment. The dataset consists of 400 images where there are 40 unique people, each having 10 images. These 40 individuals were labeled with numbers ranging for m1 to 40. This label is added as a column to the dataset.
2. The training data and test data is split based on the input of the user from the slider. By default, the value is 80. Meaning the split is going to take place on this order: 80-20.
3. Average face is calculated afterwards. This is achieved by calculating the mean by each column. The average face of the first entry of the dataset can be observed from figure 1.

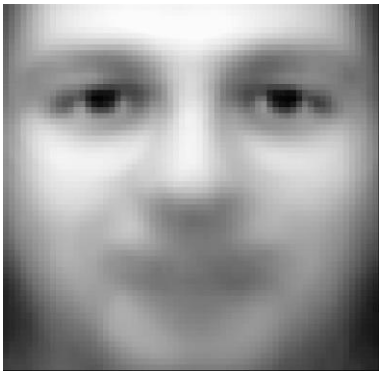

Figure 1: Average Face

4. The dataset is scaled to centered data.

5. PCA is implemented afterwards. In order to achieve this, covariance matrix of the dataset is calculated. Eigenvectors are calculated afterwards. This represents the axes of the PCA along the eigenvalues.
6. The SVD is calculated afterwards. This is used in the eigenfaces implementation and is considered to be more stable in this case. Here eigenvectors of the covariance matrix are the right singular vectors of SVD.
7. The variance ratio is calculated. This can be observed in figure 2.

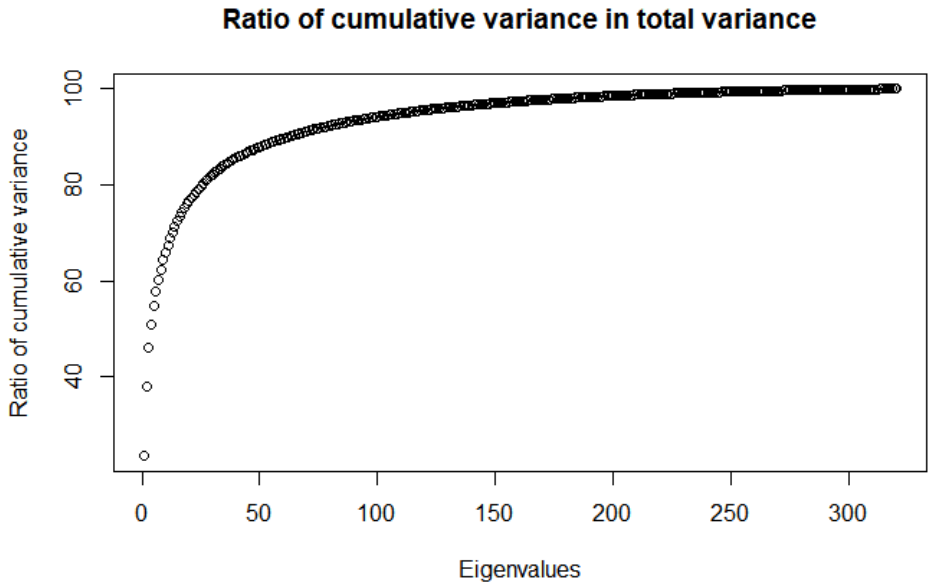

Figure 2: Cumulative variance in the total variance ratio

8. Afterwards, the principal components are selected using cumulative variance ratio with at least a total variance of 95% of the dataset.
9. Co efficient of each train face where calculated. Figure 3 shows the comparison of the training face and the reconstructed face from the eigenfaces and coefficients.

Figure 3Comparison of training and reconstructed faces

Figure 4: Eigen Faces sample

10. The matrix is projected with test data to the new reference eigenspace. The test face weight is calculated from the average train face. This result is also projected to the eigenface of the training faces. This produces a vector of weights which can be used to compare each training face weights. Distance between these two vectors helps identifying closest match.

Implementing with shiny

Shiny is an interactive framework for R. It can be used to develop user friendly dashboard or work place. The implementation has some crucial parts in terms of siny framework.

Components:

1. fileInput: Used for selecting dataset. One can browse local storage for the dataset.
2. checkboxInput: Used to be checked by user if a header existed in the dataset.
3. radioButtons: Used for two occurrences. One group of radio buttons to select the separator type of the dataset. The other one is to select the quotation type if there exists any.
4. sliderInput: This component has been used for two different occurrences. One is used to control the split of train and test dataset used in the experiment. Another slider is used to set the threshold that is used for calculating variable ration threshold to select principal components of Igenfaces.
5. textOutput: This component is used to provide labels of the images and also to show the accuracy of the classification.
6. plotOutput: This is used to plot the images generated by the code. It is to illustrate the different steps of the IgenFaces.

Flow in terms of shiny:

1. The user selects a dataset using the fileinput button.
2. After selection of the dataset, the data set is loaded in to a dataframe and passed on to the main method.
3. The main method creates several images. They are: An example of training Image, the average faces, an example of eigen faces and a reconstruction faces with coefficient and eigenvector. These images are saved on a temp directory using the png library with a resolution of 400x400.
4. The result of the PCA are stored in a data frame and stored in a csv file inside the temp directory relative to the project root directory. This is achieved with the write.csv() function.
5. Afterward the accuracy is calculated and returned. The renderText method then renders the text to the screen.
6. The renderTable and renderPlot functions are equipped with invalidateLater(1000). Which means, these functions will reload every 1 second or 1000 milliseconds. This is how the new generated images and tables are rendered to the user.