Assignment 1 - Probability and Random Processes

Ritik Dutta
16110053

20th October 2018

Face Detection and Recognition System

The following assignment involves implementing a real-time face recognition system. It is based on the following paper [1]. This method involves using features defined by a ‘“face space” which does not necessarily correspond to features we associate with human faces such as eye, ears and nose.

Mathematically, the method involves finding the principal components of the distribution of faces, i.e., the eigenvectors of the covariance matrix of the set of face images. These eigenvectors are also called as eigenfaces in the paper, since they resemble a sort of a face when displayed. To reduce the computational overhead, we choose only a few best vectors, which can account for most of the variance within the face images. These vectors are selected on the basis of the value of the corresponding eigenvalues.

All the face images in the training set can be represented as a linear combination of the eigenfaces.

The following link [2] also lists the steps which can be followed to recognise faces. The steps are listed below (Before these steps can be performed, the images containing the faces should be centered and should be of the same size):

1. Suppose an input image of dimensions $N \times N$ is stored as a $N^2 \times 1$ vector. We compute the average face vector to get the “mean face” and then subtract it from all the faces in the training set.

2. We compute the covariance matrix, given by:

$$C = A^T A$$

   where $A = [\phi_1 \phi_2 \dots \phi_M]$, M is the number of face images in the training set, ϕ is the $N^2 \times 1$ images obtained after subtracting the mean image

3. Compute the eigenvectors v_i and eigenvalues of C

4. The M eigenvalues of $A^T A$ will be the same as the M largest eigenvalues of AA^T.

5. Compute the eigenvectors of AA^T: $u_i = Av_i$ and normalise them

6. Keep the K (emperically determined) best eigenvectors

The above steps gives us a basis in the face space. We can then calculate the weights of each basis vector required to construct a given face. To recognise a new face image, we compute the weights for the new input image and for all the images in the training set. The image to which the distance of the input image is closest to is the person the input images belongs to. However, if this minimum distance is bigger than a given threshold, then the image detected is a face, but is declared as to not belong to any of the persons in the dataset.

Following are the 9 eigenfaces (eigenvectors) that were obtained with the highest eigenvalues and are used in the algorithm.

Figure 1: Eigenface 1

Figure 2: Eigenface 2

Figure 3: Eigenface 3

Figure 4: Eigenface 4

Figure 5: Eigenface 5

Figure 6: Eigenface 6

Figure 7: Eigenface 7

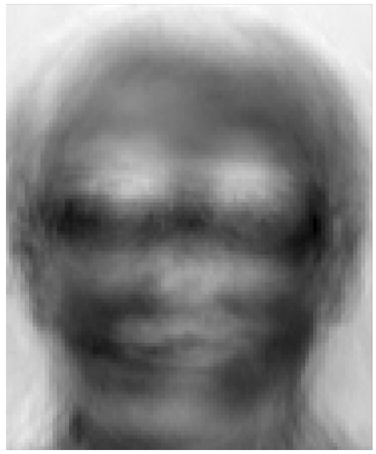

Figure 8: Eigenface 8

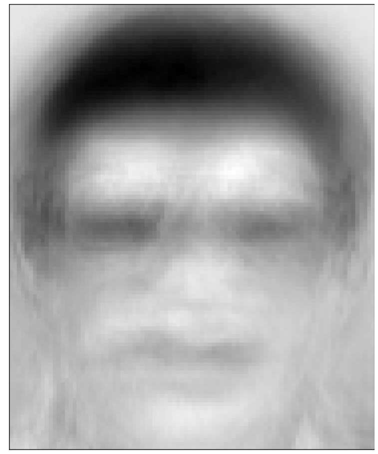

Figure 9: Eigenface 9

Results

The program was tried on three different datasets. The model is trained on the "ORL Database of Faces' dataset, which is linked [3]. 80% of the images from each class is used for training while the remaining 20% is used for testing. The accuracy on the test set was found out to be 92.5%.

Next a dataset of non-face background images was used. This dataset is linked here [4]. In this, a threshold distance of 1500 was used to separate non-face images. (ie, if the distance of the images from all the images in the test set is greater than 1500, then the image does not contain any face). The threshold was determined empirically by measuring the distance of non-face images to the images in the training set.

A different dataset of faces not belonging to the dataset used for training was also used to set the threshold for detection. The threshold was empirically set to be 1200 (ie, if the distance of the test image is less than 1200 to any of the images in the training set, then the input image has a face). However, the distance of the images in this dataset was quite close to the distances to the images in the non-face background images. (Both had distances in the range 1200-3000)

The output for images in the test set is of the following form:

Figure 10: Output

```
Predicted person:  [0] Real category:  [0]  Distance:  399.5900773401072
Predicted person:  [0] Real category:  [0]  Distance:  185.9219502119266
Predicted person:  [1] Real category:  [1]  Distance:  492.53204318881325
Predicted person:  [1] Real category:  [1]  Distance:  567.5414610883172
Predicted person:  [2] Real category:  [2]  Distance:  294.7769847723074
Predicted person:  [2] Real category:  [2]  Distance:  601.4359172963734
Predicted person:  [3] Real category:  [3]  Distance:  514.432861805718
Predicted person:  [3] Real category:  [3]  Distance:  207.24292788218472
Predicted person:  [4] Real category:  [4]  Distance:  842.7333328965697
Predicted person:  [4] Real category:  [4]  Distance:  311.7182239634707
```

The output when the images have a face:

Figure 11: Output

```
Face detected. Distance:  1853.130859683905
Face detected. Distance:  2073.732737583314
Face detected. Distance:  1646.084280579877
```

Output when the image does not contain a face:

Figure 12: Output

```
Not a face. Distance:  2334.490173414367
Not a face. Distance:  2462.825475019409
Not a face. Distance:  1806.4696856341022
```

References

1. https://ieeexplore.ieee.org/document/139758
2. http://www.vision.jhu.edu/teaching/vision08/Handouts/case_study_pca1.pdf
3. https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
4. http://www.robots.ox.ac.uk/ vgg/data/cars_brad/cars_brad.tar
5. http://www.robots.ox.ac.uk/ vgg/data/faces/faces.tar