

Machine Learning for Computer Vision Coursework 1

Face Recognition by PCA

David Angelov
MEng Electrical and Electronic Engineering
Imperial College London
david.angelov12@imperial.ac.uk

Huaqi Qiu
MSc Communication and Signal Processing
Imperial College London
h.qiu15@imperial.ac.uk

1. Question 1

In this coursework, we were given 10 normalized and vectorized face images for each of the 52 different people (referred to as 'class' in this report). Firstly, the given data set was separated into training data and testing data. In this coursework, 80% of data was used for the purpose of training and 20% of data were used for testing. The partition was done by randomly selecting 8 out of 10 images for training. Naturally, the remaining 2 images in each class were used for testing. This partition method ensured the randomness of the data sets and full coverage of classes.

Next, we applied PCA to the training data by projecting the faces to so called face space, which is spanned by M eigenfaces. To acquire the eigenfaces, we started with computing the mean face \bar{x} from the n training images x_n by

$$\bar{x} = \frac{1}{N} \sum_{n=1}^N x_n \quad (1)$$

and the resulting mean face image is shown in Figure 1.



Figure 1. Mean face

Then, the all training faces were normalized by subtracting the mean face

$$\phi_n = x_n - \bar{x} \quad (2)$$



Figure 2. The first 16 eigenfaces visualised

and the matrix A can be acquired by

$$A = [\phi_1, \phi_2, \dots, \phi_N] \quad (3)$$

Then the covariance matrix was calculated by

$$S = \frac{1}{N} A A^T \quad (4)$$

The eigenvectors u_i were calculated from the covariance matrix S by $S u_i = \lambda_i u_i$. The dimension of the training data vector x_n is $D = 2576$, denoted by $x_n \in R^D$ (for the data provided for this coursework). To reduce the dimension and maximising the variance of projected data, we selected M of the eigenvectors corresponding to the M largest eigenvalues λ_i as eigenfaces. We found that among all the λ_i ($i = 1, 2, \dots, D$), 415 out of D eigenvalues were considered large (or non-zero), with order of magnitude ranging from 1 to 5. The remaining $D - 415 = 2161$ of the eigenvalues were considered as zeros with order of magnitude ranging from -10 to -13. In this coursework, we selected the largest $M = 50$ eigenfaces as the basis vectors for the face space. In Figure 2, we visualised 16 out of these 50 eigenfaces. The eigenfaces were visualised by reshaping the x_n vector to a 56×46 matrix.

To apply the PCA method, we projected every normalised training face ϕ_n to the M -dimensional subspace. Namely,

$$\omega_n = [a_{n1}, a_{n2}, \dots, a_{nM}] \quad (5)$$

where the projection of n th image on the i th eigenface is denoted by

$$a_{ni} = \phi_n^T \mathbf{u}_i, \quad i = 1, \dots, M \quad (6)$$

Thus, the training faces were represented by its projections ω_n .

2. Question 2

In this question, we discuss two different methods of calculating covariance matrix S in the scope of PCA. It can be noticed from Eq.4 that the dimension of the covariance matrix is $D \times D$, which equals to the number of pixels in an image and is typically large. Under the consideration of computational efficiency, we implemented another proposed method, in which the covariance matrix is calculated by

$$S = \frac{1}{N} A^T A \quad (7)$$

Since $A \in R^{D \times N}$, the covariance matrix now has dimension of $N \times N$, where N equals to the number of images and is typically much smaller than D . As computing eigenvectors of large matrices is computational expensive, this method is preferred if proved equally effective for PCA. The resulting non-zero eigenvalues from Eq.7 had the same value of that using Eq.4. The number of non-zero eigenvalues in this case is also 415. Note that the eigenvectors using this method were converted to the D -dimensional eigenvectors by

$$\mathbf{u}_i = A \mathbf{v}_i \quad (8)$$

where \mathbf{v}_i denotes the eigenvectors calculated using the covariance matrix in Eq.7. Figure 3 shows the visualised 16 eigenfaces. Comparing to the eigenfaces produced by the previous method (Figure 2), we notice that the face contour of the eigenfaces are the same. The difference in grey scale value could be caused by the randomisation of the data set and automatic normalisation by the image display function `imagesc` in MATLAB.

In conclusion, the methods for the computation of covariance matrix in Q1 and Q2 were proven equally effective for PCA, while the latter method is less computational expensive. Hence we used $S = (1/N)A^T A$ to compute the covariance matrix for image recognition in later sections of this coursework.

3. Question 3

The face images can be reconstructed from their projection on the M -dimensional subspace ω_n by



Figure 3. The first 16 eigenfaces visualised for question 2

$$\tilde{\mathbf{x}}_n = \bar{\mathbf{x}} + \sum_{i=1}^M a_{ni} \mathbf{u}_i \quad (9)$$

where $\tilde{\mathbf{x}}_n$ is the reconstructed face image. The reconstruction error, denoted as J can be evaluated by

$$J = \frac{1}{N} \sum_{n=1}^N \|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2 \quad (10)$$

where N is the total number of images. The formulation of PCA that we applied in this coursework maximises the variance of the projection of training data by spanning the face space using M eigenvectors $\{\mathbf{u}_i\}$ of S corresponding to the largest M eigenvalues $\{\lambda_i\}$ where $(i = 1, 2, \dots, M)$. This formulation equivalently minimises the reconstruction error J so it theoretically becomes

$$J_{theo} = \sum_{i=M+1}^{D^*} \lambda_i \quad (11)$$

The D^* in Eq.11 is the total number of eigenvectors of S . For the method we selected in Question 2, $D^* = 416$, which equals to the total number of training images. We evaluated and compared the experimental J calculated by Eq. 10 and theoretical J_{theo} calculated by Eq.11 while varying the number of M . Figure 4 presents J and J_{theo} as M varying from 40 to 100.

It can be noticed that 1) the theoretical and experimental reconstruction errors are identical for all value of M ; 2) the reconstruction error reduces when the number of M increases, but the gradient of the decreasing error curve reduces. This indicates that the positive effect of using more

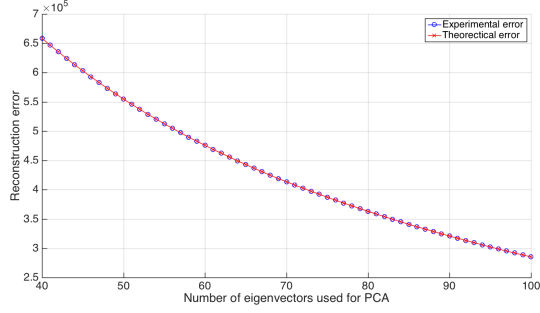


Figure 4. Reconstruction error, experimental vs theoretical

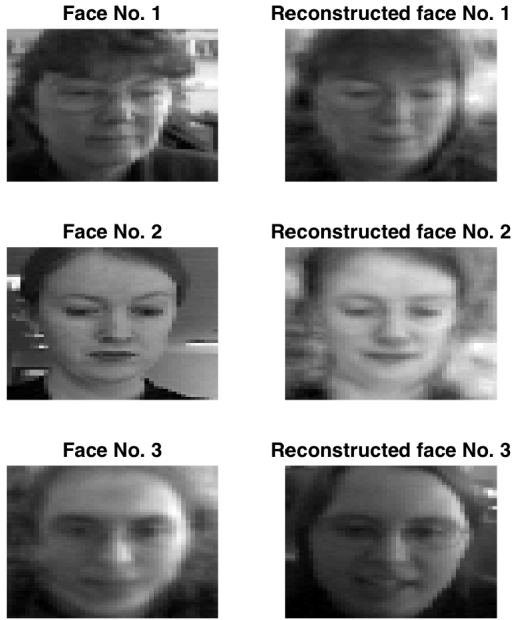


Figure 5. Visualise reconstructed training images

number of eigenfaces on reducing J decreases as M increases.

Figure 5 shows the original training faces and the corresponding reconstructed images. Similarly, Figure 6 shows the original testing faces and corresponding reconstructed images. As we can observe, the face images were well-reconstructed for both training and testing faces.

4. Question 4

At the training stage, the training images were projected on the M -dimensional face space. The n^{th} image in the training set was represented as ω_n . At testing stage, each testing image was normalised by Eq.2 and then projected

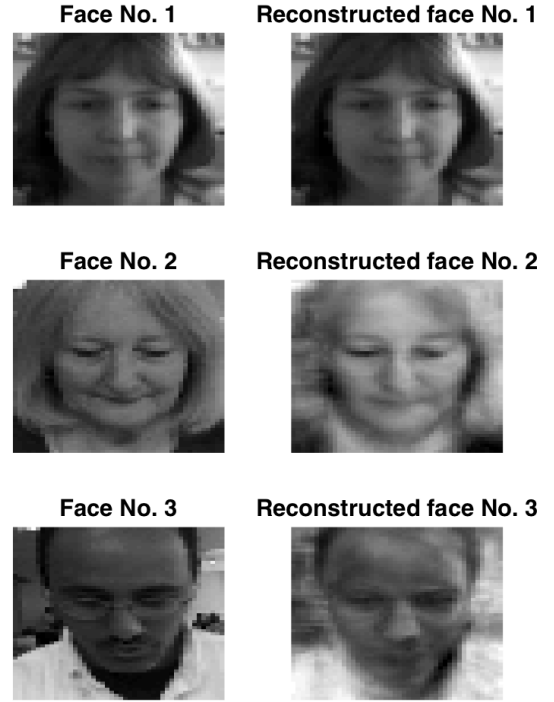


Figure 6. Visualise reconstructed testing images

on the eigenspace by Eq.6 and represented as

$$\omega = [a_1, a_2, \dots, a_M]^T$$

This process transformed the testing images to represented data points that can be classified to a specific image in training set, thus effectively, recognised. In this coursework, the classification was achieved by Nearest Neighbour method and Alternative method. We shall now discuss and compare the two methods.

4.1. Nearest Neighbour classification

In the Nearest Neighbour (NN) classification, the testing image is assigned to the class which has the minimum error e , which is calculated with respect to all training images as

$$e = \|\omega - \omega_n\|, n = 1, \dots, N \quad (12)$$

where recall N is the number of training images.

4.1.1 Accuracy of recognition

Each of the testing image was assigned a label indicating which class the image was classified to. This is referred

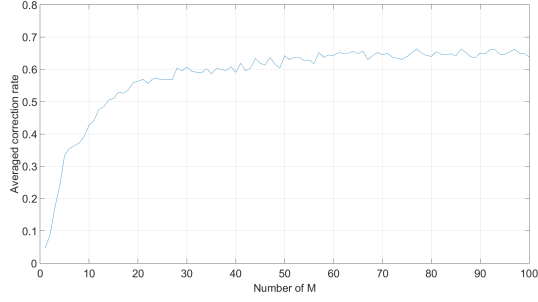


Figure 7. Correction rate of NN method

to as prediction in the report. To evaluate the accuracy of recognition, we calculated correction rate by

$$\text{correction rate} = \frac{\text{number of correct prediction}}{\text{number of testing image}} \quad (13)$$

The correction rate of PCA based face recognition can be affected by the number of eigenvector used to span the face subspace M . Hence, we varied the number of M from 1 to 100 and calculated the corresponding correction rate. Considering the randomness of data partition, we iterated the traing-testing process for 50 times for each value of M and averaged the correction rate. The result of the calculations are presented in Figure 7. The correction rate increased rapidly as M increased from 1 to around 30 and then converged to approximately 65%. This indicates that it is unnecessary to use large number of eigenvector basis u_i , despite there are 415 of u_i corresponding to largest eigenvalues available. As the computational complexity increases when M increases, we came to the conclusion that a optimal value of M is around 50-60 for this particular situation, as the correction rate started to converge in this region.

4.1.2 Example of failure cases

We now discuss the cases of failure for NN method classification. Two examples of failed classifications are shown in image (a) and (c) in Figure 8. In this figure, image (b) and (d) are the images from the classes that image (a) and (c) were classified to in testing. In both comparison between both pairs (a)-(b) and (c)-(d), we can observe some non-decisive similarities in terms of facial feature. Similar comparisons were made on other misclassified testing images. However, no clear visual indication of the cause of incorrect recognition was established.

4.1.3 Confusion matrix

The confusion matrix provides a visual impression of the recognition result. The confusion matrix of PCA based face



Figure 8. (a)&(c) Image of failed cases for NN method; (b)&(d) sample image from the class (a)&(c) are classified as

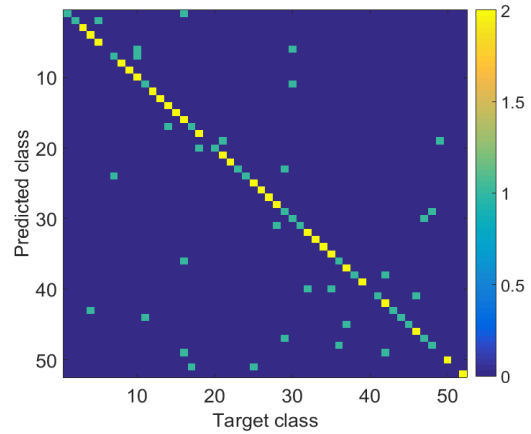


Figure 9. Confusion matrix of NN classification, $M = 100$

recognition using NN classification is presented in Figure 9, which corresponds to a correction rate of 63%. In the confusion matrix, the colour patches on the diagonal indicate correct predictions. Correspondingly, patches appeared on places other than the diagonal indicate a case of incorrect classification and the number of misclassified cases are indicated by the colour.

4.2. Alternative method for classification

An alternative method oppose to the NN method was implemented in our coursework for classification of data points ω . This method computes the eigenvectors and creates face subspace for every class separately and store the subspaces as training knowledge. At testing stage, the testing image was projected to individual subspaces using the Eq.6 and reconstructed using Eq.9. The assignment of predicted label depends on the measurement of reconstruction

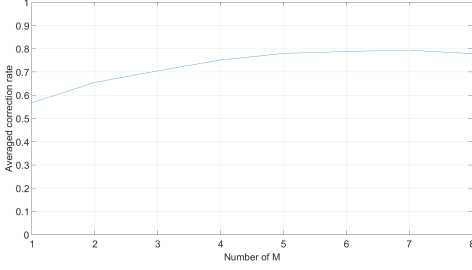


Figure 10. Correction rate for the alternative method

error by

$$\hat{y} = \arg_i \min \|\mathbf{x} - \tilde{\mathbf{x}}_i\| \quad (14)$$

where \mathbf{x} is the testing image and $\tilde{\mathbf{x}}_i$ is the reconstructed testing image using the eigenvector subspace computed from the i th class.

4.2.1 Accuracy of recognition

We tested all available testing images and calculate the correction rate as a measure of recognition accuracy. The correction rate is calculated using the same method used in NN method (Eq.13). Similarly, we varied the number of eigenvectors used in each class to span the face subspace for the classes. Thus the number of eigenvector basis M_{alter} can be varied between 1 and 8 and the total number of eigenvectors used is $52 \cdot M_{alter}$. Figure 10 is the resulting correction rate plotted against the value of M_{alter} . Same as in NN method, the correction rate was averaged over 50 iterations for every value of M_{alter} . The result indicates that 1) the alternative method achieved better correction rate than the NN method; 2) the conclusion regarding the impact of number of eigenvector used to the accuracy drawn from the result of the NN method is also valid.

4.2.2 Example of failure cases

Here in Figure 11, we demonstrate the cases of failed recognition as in NN method to visually exam the misclassification. Again we can observe some suspicious cause of misclassification such as the similarity of lighting in between (a) and (b), or the similarity of the shape of forehead between (c) and (d). However, these observations are rather intuitive and subjective and thus cannot be related to the condition that caused misclassification. The fact that the significant features (i.e. eigenfaces) extracted for PCA based recognition are not necessarily related with typical visual features (e.g. eyes, nose, mouth etc.), visual examination of failure cases may not be a constructive effort to identify the cause of failure.

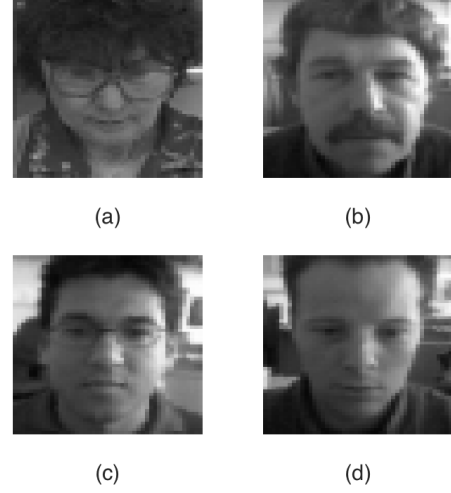


Figure 11. Failed cases for the alternative method

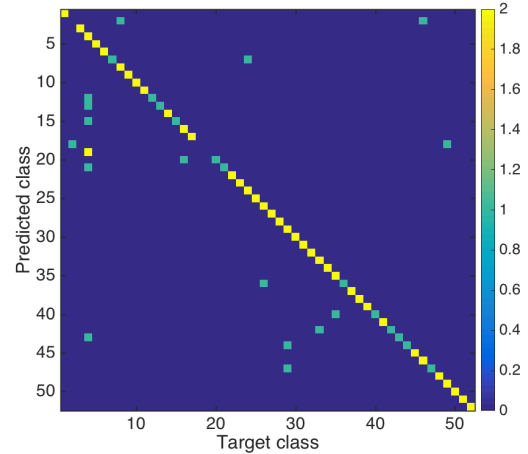


Figure 12. Confusion matrix of alternative method classification, $M_{alter} = 8$

4.2.3 Confusion matrix

Figure 12 show the confusion matrix of alternative method of classification, which corresponds to a correction rate of 78%.

4.3. Comparison between NN and alternative method

1. Accuracy

As we can see from Figure 7 and 10, the alternative method achieved higher accuracy in general. However, it's worth mentioning that the minimum amount of eigenvectors required for the alternative method is 52

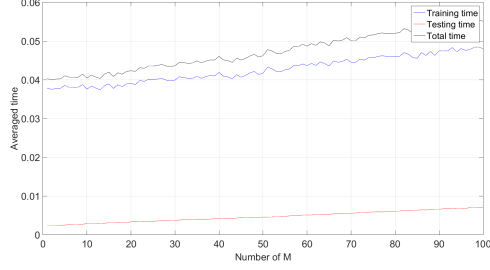


Figure 13. Time consumed for the NN method

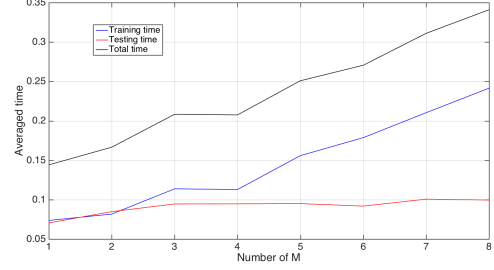


Figure 14. Time consumed for the alternative method

(i.e. $M_{alter} = 1$). The accuracy rate achieved at this number of M_{alter} is lower than the accuracy of NN method with ($M = 52$). Other than this exception, the alternative method delivered better recognition accuracy.

2. Time consumption

To compare the two classification methods in terms of time efficiency, we measured the time consumed by the training and testing program by the stopwatch in MATLAB `tic` and `toc`. Again, we varied the number of eigenvectors used to measure the effect of that too. Figure 13 and 14 show the training time, testing time and total time of NN method and the alternative method respectively. It can be concluded from the results that: a) The training time contributed to the majority of total time consumption for both methods; b) The time required for the NN method did not increase with increasing number of M , while the training time of the alternative method had a noticeable increase. This conclusion in particular reveals a trade-off between time consumption and accuracy for the alternative method when selecting the value of M_{alter} ; c) the alternative method appears to be more time consuming than the NN method in our measurement. However, the difference between the two methods when the same number of eigenvector were used was multiple of 10ms and could be subjective to the optimisation of programming.

3. Memory

If we assume that the same amount of eigenvectors (M) were used for both methods, it can be found that both the NN method and the alternative method requires to store data of size $[M \times D]$ ($M = M_{alter} \times 52$ for the alternative method). Consequentially, the memory requirement of the two classification methods is primarily affected by the number and size of eigenvectors, as they are also most memory demanding and cannot be released during both training and testing period. However, it can be observed from Figure 7 and

10, the NN method requires fewer eigenvectors, equivalently fewer memory, to achieve its maximum accuracy.

[1]

References

- [1] T. K. Kim. Face recognition by eigenfaces: Principle component analysis, 2016. Lecture notes of Machine Learning for Computer Vision.