Assignment 2 : Eigenfaces for Face Recognition

Loïc Tetrel

McGill University
845 Rue Sherbrooke O, QC H3A 0G4, Montréal, Canada
loic.tetrel@mail.mcgill.ca

1 Introduction

The use of principal component analysis has been widely use in the area of computer vision. In this assignment we will investigate it use for face recognition. Because the dimension of such point is really big (393216 dimensions point for image 512×768), the main idea is to project the image into a sublower space called "face-space" that best encodes the variation among known face images [2] and it is derived from the Fourier transform. Each M image characterized by $[1, N^2]$ with N^2 dimensions differs from the average $\Psi = \frac{1}{M}\sum_{n=1}^{M} \Gamma_n$ by the matrix $A = \Gamma - \Psi$. Principal component analysis will find the M orthogonal vectors v_M that maximizes the variance λ_k along each vector, i.e. we search the eigen vectors v_k and eigen values λ_k of the covariance matrix C :

$$C = A^t A \tag{1}$$

Because the covariance C is N^2 by N^2, the snapshot method [3] says that the best λ_k are the same as $C = AA^t$ which is M by M. Usually the training set is much lower than the number of dimension ($M << N^2$) so this allowed us to compute v_M and λ_k much faster. The mapping from the pixel-space to the face-space and is given by :

$$u = v^t A \tag{2}$$

The matrix u is called the eigen faces and is M by N^2.

2 Principal component analysis for subject classification

The FERET database [1] is used in this assignment. The pose nomenclature is wrong and all the r* will have reverse angles values, re is +80°. I will use 28 images for all the 52 subjects for the training, and 4 images for the testing.

2.1 Results of principal component analysis

The mean image of the FERET are shown in Fig.1.Looking at the eigen faces (Fig. 2), it can be seen that the first eigen vector looks like a human face. However there is no information about the hairs, represented by u_5 or u_8. The last faces are very dark and there isn't a lot of information coming from it.

Fig. 1: Mean face of the training set.

Fig. 2: The 10 first eigen spaces.

2.2 Reconstruction

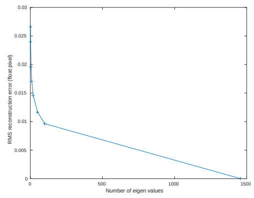
One way to evaluate the number of dimensions used in our new space is to (re)project the images in the pixel space using a fewer dimension. Then the reconstruction error can be computed between the truth image Γ_t and the projected image Γ_p :

$$Rec_{error} = \sum_{M}\left(\sqrt{\sum_{N^2}(\Gamma_t - \Gamma_p)^2}\right) \tag{3}$$

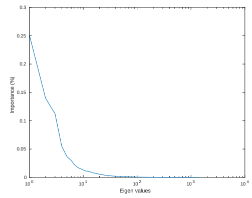
Obviously, the more we use eigen values, the more accurate we are. Taking into account the future computational time, it is more interesting to take less eigen values like 50 which give an accuracy of 1% not far from the standard 5% (Fig. 3). It is also interesting to see the importance of each eigen values. The Figure 4 shows qualitatively the accuracy for 3 random images from the training set with $1, 5, 10, 20, 50$ eigen values, and we see that using just a few eigen faces does not produce good results.

2.3 Classification and recognition

The classification with eigen faces can be done using a nearest neighbour algorithm. The classification for one image Γ is the label k if the nearest image of Γ is a subject k. The distance between images $d_{\Gamma_{ij}}$ is define as the norm of the eigen weights $w = Au'$.



(a)



(b)

Fig. 3: (a) RMS error reconstruction versus number of eigen values. (b) Importance of the eigen values.

Fig. 4: RMS error reconstruction versus number of eigen values.

$$d_{\Gamma_{ij}} = \|w_i - w_j\| \tag{4}$$

The error decrease with the number of eigen values. The optimal number of eigen values seem to be at minimum 20, because the error start to be constant at 30% with this value 5. The error min is given for 50 eigen values.

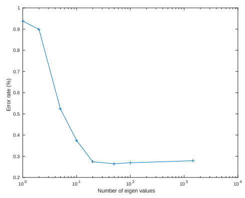


Fig. 5: Error recognition rate.

The Figure 6 shows three missclassified images and three good classification. We wee that even for a human the missclassified images are not obvious.

Fig. 6: Missclassified and good classified examples.

2.4 Probabilistic Face Recognition

The classification can be also computed using a bayesian rule. Assuming that the data d (pixels) are well fitted to the model m (eigen values), the probability for one class l given an image I is :

$$p(l = k|I) = \frac{p(I|l = k)\ p(l = k)}{p(I)} \tag{5}$$

Because $p(l = k|I)$ α $p(I|l = k)p(l = k)$, to assign a label to a given image we will assume the following assumption :

$$\max p(l = k|I) \simeq \max p(I|l = k) \tag{6}$$

The likelihood $p(I|l = k)$ for each subject k will be approximated by a multivariate gaussian on 28 images for each subject with 50 eigen weights with the covariance Σ 50 by 50 and mean μ. Using 50 values, the recognition rate is 60% which is lower than 75% for the nearest neighbour and I think this is due to the number of the training. When we take not enough training data, the eigen weights are not well distributed and there are computation problem with the covariance. The Figure 7 shows the distribution of the previous images of Figure 6.

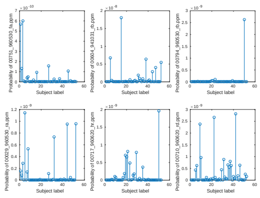


Fig. 7: Likelihoods of missclassified and good classified examples.

3 Principal component analysis for pose classification

In this section, we want to classify the pose of each subject. We will map poses to seven coarse pose class $\Theta = [-90°, -50°, -15°, 0°, +15°, +50° + 90°]$. 90% of the data will be used for training and 10% will be used for testing.

3.1 Results of principal component analysis for pose classification

The mean image of each pose class are shown in Fig.8. The 10 first eigen faces for each pose class are shown in Figure 9. We can see that the eigen faces follows well the angles.

3.2 Pose Classification

Using the same method that in the previous section, we will classify each testing image using nearest neighbourhood. We have to project each testing image into 7 different space each representing one pose. We compute the nearest neighbour for one space and we take the distance value for the nearest image. Then we have 7 possible images (each the nearest for one pose space), the minimun distance of these 7 images is our class prediction. The marginal error rate for with 1, 2,

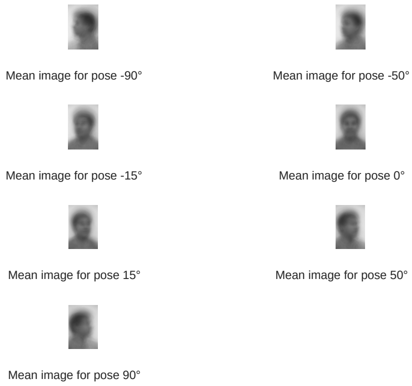


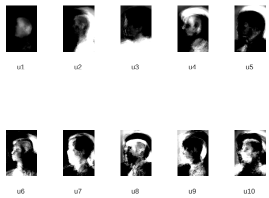
Fig. 8: Mean face of each pose of the training set.

5, 10, 20, 30, 50, 90, 100, all eigen values is shown in Figure 10. The error rate is quite high because the algorithm tend to confound people when there are not the same and in the same orientation (Fig. 11). All the confusion matrices are shown in Table 1-10. It can be seen that the more we take eigen values, more accurate we are. Many predicted poses are near the diagonal, that mean that the algorithm confound near pose (Fig. 11), but recognize well when the two images has more distinct poses.

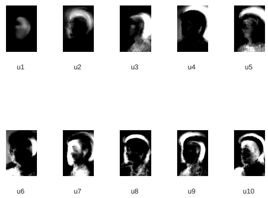
Table 1: Confusion matrix for 1 eigen value.

Prediction	Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6	Θ_7
Θ_1	3	2	1	2	2	4	2
Θ_2	2	3	1	2	5	5	5
Θ_3	0	3	3	7	3	5	1
Θ_4	0	2	10	4	6	6	4
Θ_5	3	2	2	6	6	5	3
Θ_6	1	1	3	4	5	5	3
Θ_7	1	3	2	4	5	4	4
Actual							

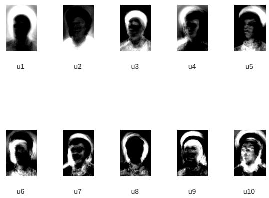
The Figure 11 shows three missclassified images and three good classification. We wee that when the training image has the same face in testing it's easier for the algorithm to find the good position.



(a) $\Theta_1 = -90^\circ$



(b) $\Theta_2 = -50^\circ$



(c) $\Theta_3 = -15^\circ$

(d) $\Theta_4 = 0^\circ$

(e) $\Theta_5 = +15^\circ$

(f) $\Theta_6 = +50^\circ$

(g) $\Theta_7 = +90^\circ$

Fig. 9: Top 10 eigen faces u_i for each pose class Θ_j.

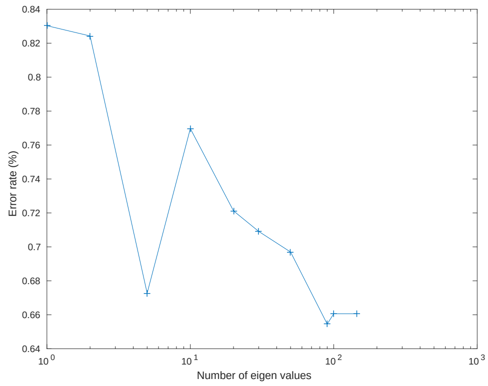


Fig. 10: Error rate for eigen values. Nota : the last number of eigen values depends on the training pose set because it is not uniform, so I took an average of 150.

Table 2: Confusion matrix for 2 eigen value.

Prediction	Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6	Θ_7
Θ_1	4	1	0	2	4	4	1
Θ_2	2	3	4	5	5	4	0
Θ_3	0	1	5	6	3	5	2
Θ_4	0	4	6	6	10	3	3
Θ_5	0	1	4	7	4	6	5
Θ_6	0	2	2	2	6	3	7
Θ_7	0	5	2	5	4	3	4
Actual							

Table 3: Confusion matrix for 5 eigen value.

Prediction	Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6	Θ_7
Θ_1	4	7	2	2	0	1	0
Θ_2	4	10	4	4	1	0	0
Θ_3	0	2	7	8	4	1	0
Θ_4	2	0	6	10	10	2	2
Θ_5	0	0	1	13	7	5	1
Θ_6	0	0	2	2	4	5	9
Θ_7	0	0	1	2	0	9	11
Actual							

Table 4: Confusion matrix for 10 eigen value.

Prediction	Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6	Θ_7
Θ_1	1	9	4	1	0	1	0
Θ_2	6	6	7	2	1	1	0
Θ_3	0	3	9	7	3	0	0
Θ_4	0	1	9	8	12	1	1
Θ_5	0	1	0	13	7	5	1
Θ_6	0	0	1	1	8	2	10
Θ_7	0	0	0	2	1	15	5
Actual							

Table 5: Confusion matrix for 20 eigen value.

Prediction	Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6	Θ_7
Θ_1	4	9	1	0	2	0	0
Θ_2	8	5	5	4	1	0	0
Θ_3	0	3	8	7	4	0	0
Θ_4	0	0	6	9	15	1	1
Θ_5	0	0	2	13	8	4	0
Θ_6	0	0	1	1	8	2	10
Θ_7	0	1	0	1	0	11	10
Actual							

Table 6: Confusion matrix for 30 eigen value.

Prediction	Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6	Θ_7
Θ_1	4	9	1	0	2	0	0
Θ_2	9	6	5	3	0	0	0
Θ_3	0	3	8	7	4	0	0
Θ_4	0	0	7	8	15	1	1
Θ_5	0	0	2	12	8	5	0
Θ_6	0	0	0	1	9	2	10
Θ_7	0	0	1	0	1	9	12
Actual							

Table 7: Confusion matrix for 50 eigen value.

Prediction	Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6	Θ_7
Θ_1	5	9	1	0	1	0	0
Θ_2	10	6	4	3	0	0	0
Θ_3	0	3	9	6	4	0	0
Θ_4	0	0	6	9	15	2	0
Θ_5	0	0	1	14	8	4	0
Θ_6	0	0	0	1	9	2	10
Θ_7	0	0	1	1	0	10	11
Actual							

Table 8: Confusion matrix for 90 eigen value.

Prediction	Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6	Θ_7
Θ_1	5	9	1	0	1	0	0
Θ_2	9	8	4	2	0	0	0
Θ_3	0	3	10	6	3	0	0
Θ_4	0	0	6	12	13	1	0
Θ_5	0	0	0	14	9	4	0
Θ_6	0	0	0	1	9	3	9
Θ_7	0	0	0	2	0	11	10
Actual							

Table 9: Confusion matrix for 100 eigen value.

Prediction	Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6	Θ_7
Θ_1	5	9	1	0	1	0	0
Θ_2	9	8	4	2	0	0	0
Θ_3	0	3	10	6	3	0	0
Θ_4	0	0	6	12	13	1	0
Θ_5	0	0	1	14	8	4	0
Θ_6	0	0	0	1	9	3	9
Θ_7	0	0	0	2	0	11	10
Actual							

Table 10: Confusion matrix for all eigen value.

Prediction	Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6	Θ_7
Θ_1	5	9	1	0	1	0	0
Θ_2	11	7	4	0	0	0	1
Θ_3	0	3	11	4	4	0	0
Θ_4	0	0	10	9	12	1	0
Θ_5	0	0	1	13	9	4	0
Θ_6	0	0	0	1	9	3	9
Θ_7	0	0	0	0	0	11	12
Actual							

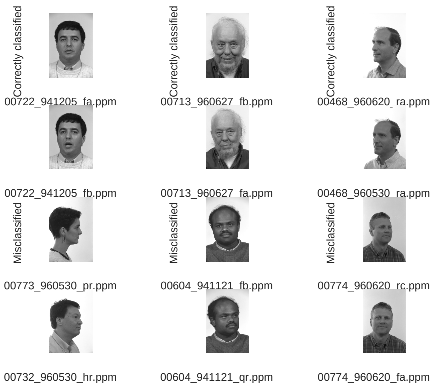


Fig. 11: Missclassified and good pose classified examples.

3.3 Probabilistic Pose Classification

The classification can be also computed using a bayesian rule. Assuming that the data d (pixels) are well fitted to the model m (eigen values), the probability for one class l given an image I is :

$$p(l = k|I) = \frac{p(I|l = k)\ p(l = k)}{p(I)} \tag{7}$$

Because $p(l = k|I)\ \alpha\ p(I|l = k)p(l = k)$, to assign a label to a given image we will assume the following assumption :

$$\max p(l = k|I) \simeq \max p(I|l = k) \tag{8}$$

The likelihood $p(I|l = k)$ for each subject k will be approximated by a multivariate gaussian on all images for each pose with 50 eigen weights with the covariance Σ 50 by 50 and mean μ. Using 50 values, the recognition rate is 27% (Tab. 11) which is lower than 35% for the nearest neighbour. When we take not enough training data, the eigen weights are not well distributed and there are computation problem with the covariance. The Figure 12 shows the distribution of the previous images and we see that it tends to be uniform.

Table 11: Confusion matrix for 50 eigen value.

Prediction	Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6	Θ_7
Θ_1	7	3	2	0	0	1	3
Θ_2	8	5	3	7	0	0	0
Θ_3	6	3	4	8	0	0	1
Θ_4	12	1	3	10	4	0	2
Θ_5	10	1	1	3	3	5	4
Θ_6	7	0	4	1	5	4	1
Θ_7	3	0	3	0	6	0	11
Actual							

4 Conclusion

The goal of this assignment was to perform image classification using PCA. We saw two approaches on two problems : Nearest neighbour and Bayes with face and pose classification. The neighbouring approach does not need assumptions in the data, however it just give us a strict label without more information. In the other hand, the bayesian approach need one assumption and a multivariate gaussian assumption was made. It is difficult to modelise the likelihoods of the subjects if there is no precision about the underlying model, but it has the advantage to give more information than NN.

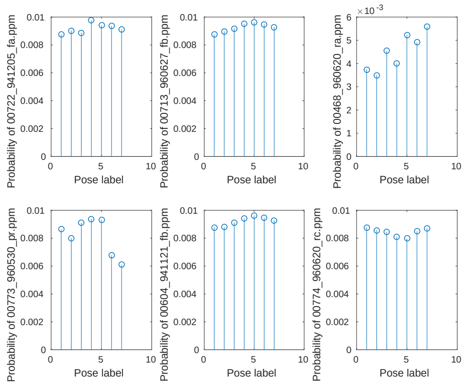


Fig. 12: Likelihoods of missclassified and good pose classified examples.

Face classification gives better recognition rate ($\simeq$ 70%) than pose recognition ($\simeq$ 35%). I explain this because the eigen faces are more sensitive to the human face, than to its position. The PCA has difficulties to be consistent in general, here with angles but also with lighting condition or scale.

References

1. Phillips, P.J., Wechsler, H., Huang, J., Rauss, P.J.: The feret database and evaluation procedure for face-recognition algorithms. Image and vision computing 16(5), 295–306 (1998)
2. Sirovich, L., Kirby, M.: Low-dimensional procedure for the characterization of human faces. Josa a 4(3), 519–524 (1987)
3. Turk, M.A., Pentland, A.P.: Face recognition using eigenfaces. In: Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on. pp. 586–591. IEEE (1991)