ECSE-626 STATISTICAL METHODS IN COMPUTER VISION

Assignment #2

Due: Friday, March 18, 2016, 11:59pm

Please submit your assignment solutions electronically via the myCourses assignment dropbox. The solutions should be in PDF format. Attempt all parts of this assignment. This assignment is out of a total of **100 points**. It is designed to familiarize the students with Principle Components Analysis (PCA) in the context of image reconstruction and recognition. The assignment requires the students to use Matlab to create and evaluate Eigenspace representation of facial images in face recognition and pose estimation.

Some questions require writing code in Matlab. The submitted code should have separate local functions to perform the required functionalities for each part of the assignment. A set of images is included with this assignment. Examples of code can be readily obtained from the internet, however the student is expected to write his/her own code.

Assignments received up to 24 hours late will be penalized by 30%. Assignments received more than 24 hours late will not be marked.

Face dataset

For this assignment, we use face images from a subset of the publicly available Color FERET Database. You can download a zipped archive of this subset, titled "Color FERET Database.7z", from myCourses. In this folder, there are 52 directories, each containing images of a specific subject in different poses (the Color FERET database contains images from approximately 1000 different subjects). In the given subset, the number of images per subject varies from 32 to 96. Each image is 768×512 pixels and the files are in PPM-format. The file names are in the form of nnnnn_yymmdd_xx.ppm, where nnnnn is the subject ID, yymmdd is the capture date and xx is the FERET pose name, shown in Table 1.

Table 1: FERET pose names (consider the pose names to lowercase letters only). [From http://www.itl.nist.gov/iad/humanid/feret/feret_master.html]

Two letter code	*Pose Angle (degrees)*
Fa	0 = frontal
Fb	0
ba	0
bj	0
bk	0
bb	+60
bc	+40
bd	+25
be	+15
bf	-15
bg	-25
Bh	-40
bi	-60
ql	-22.5
qr	+22.5
hl	-67.5
hr	+67.5
pl	-90
pr	+90
Ra	+45
Rb	+10
Rc	-10
Rd	-45

1 Eigenfaces for Face Recognition

In this part, we create an Eigen space using a training set of facial images from each subject. Since we want to do face recognition, we will use equal number of training images for each subject. Otherwise, the method will be biased towards the subjects with more training images.

To create the training and testing datasets, randomly select 28 images per subject for the training set and randomly select 4 images per subject for the testing set (each subject is guaranteed to have at least 32 images). Note that the training and the testing sets should not have any images in common. Since the resolution of the images are rather high, you advised to resize both the training and the testing images by the factor of 0.5. You should also convert the images to gray scale.

1.1 Principal Components Analysis

Write a program which

- Randomly selects 28 training images from each subject to form the training set
- Randomly selects 4 testing images from each subject to form the testing set
- Reads in all the training images
- Creates an $N \times D$ data matrix, each row containing a training image
- Computes the mean image of the training set
- Subtracts the mean image from each training image
- Computes principal components of the resulting data using the snapshot method

In your report

1. Show the mean image for this data set. (5 point)
2. Show the first 10 Eigenfaces. (5 point)
3. Discuss how many principal components are required to represent the data based on your observation of the principal components. (1 point)

1.2 Reconstruction

One way to evaluate the performance of Eigenfaces to represent face images is to project a face image onto the Eigen space and then reconstruct the face image by projecting back from the Eigen space.

Write a program which

- Projects all the training images onto the Eigen space, reconstructs them, and computes the reconstruction error for the following number of eigenvectors: $\{1,2,5,10,20,50,100, all\}$.

In your report

4. Plot the average reconstruction error versus the number of Eigenfaces. Based on your observation, what is the optimal number of Eigenfaces? (5 point)
5. Randomly choose three images from the training set. Display their reconstructed versions for the following number of eigenvectors $\{1,5,10,20,50\}$. (5 point)

1.3 Classification and recognition

Write a program that for the following number of eigenvectors $\{1,2,5,10,20,50,100, all\}$,

- Finds the Eigenspace representation of all the training images
- Finds the Eigenspace representation of all the testing images
- Classifies the subject label of the test images based on the Euclidean nearest neighbor rule in the Eigenspace (i.e., finds the training sample with the closest Eigenspace representation in terms of Euclidean distance).
- Assigns the subject label of the test image equal to the subject label of the nearest training image
- Calculates the recognition rate of the method as the ratio of the correct labels to the total number of test images

In your report

6. Plot the average recognition rate versus the number of Eigenfaces for the following number of eigenvectors: $\{1,2,5,10,20,50,100, all\}$. Based on your observation, what is the optimal number of Eigenfaces? (10 point)
7. Display the nearest training image for three correctly classified and three misclassified test image, using the first 50 eigenvectors. (5 point)

1.4 Probabilistic Face Recognition

Consider the task of face recognition. Let $l \in \{1 \dots K\}$ denote the subject label, where K is the number of subjects in the training set and let I be an image projected into the Eigenspace.

8. Using the Bayes rule, derive an expression for the posterior density of a subject label, given an image $p(l = k|I), k \in \{1 \dots K\}$. (2 point)

To perform probabilistic recognition, write a program which does the following using the first 50 eigenvectors:

- Finds the Eigenspace representation of all the training images
- Assuming Gaussian distributions, calculates the likelihoods for each subject
- Finds the Eigenspace representation of all the testing images
- Computes the posterior distribution of the subject label for each testing image
- Uses MAP to assign the subject label of the test images
- Calculates the recognition rate of the method as the ratio of the correct labels to the total number of test images

In your report

9. Compare the recognition rate of the probabilistic face recognition with the nearest neighbor classifier, each using the first 50 eigenvectors. (10 point)
10. Display the posterior distribution of the subject label for the same images that you used in Question 7. (5 point)

2 Eigenfaces for Pose Estimation

Since the pose of the images in the Color FERET Database are labeled, we can use them to create separate Eigenspace representations and use them to classify the pose of an unseen image. As shown in Table 1, the pose labels belong to the following set: $\{0, \pm 10, \pm 15, \pm 22.5, \pm 25, \pm 40, \pm 45, \pm 60, \pm 67.5, \pm 90\}$. We can further map them into the following set of seven coarse pose classes: $\Theta = \{0, \pm 15, \pm 50, \pm 90\}$.

2.1 Eigenspace for each Pose Class

Write a program which does the following to create an Eigenspace for each pose class θ_i

- Finds all the images from all subjects belonging to θ_i (use the file name of each image to determine its FERET pose name and map them to the closest category in Θ)
- Randomly chooses 90% of the images for training and the other 10% for testing
- Creates the Eigen space using the training set

In your report

11. Display the mean image of each pose class. (10 points)
12. Display the first 10 Eigenfaces for each pose class. (5 point)

2.2 Pose Classification

Write a program which

- Finds the Eigenspace representation of the training images of each pose class (use the corresponding Eigenspace)

- Finds an Eigenspace representation of each testing image for each pose class (use the corresponding Eigenspace)
- Classifies the pose of the test image based on the Euclidean nearest neighbor (search for the nearest neighbor among all the Eigenspaces)
- Assigns the pose class of the test image equal to the pose class of the nearest training image
- Calculates the confusion matrix for pose classification

Since we have seven different pose classes, the confusion matrix for pose classification would be a 7×7 matrix, with its ijth element illustrating the normalized number of times the ith pose is classified to the jth pose. Note that each row of the matrix sums to one. Note that a perfect pose classification would result in a diagonal confusion matrix.

In your report

13. Plot the confusion matrix for pose classification for the following number of eigenvectors: $\{1,2,5,10,20,30,50,90,100, all\}$. (10 point)
14. Display the nearest training image for three correctly classified and three misclassified test image, using the first 50 eigenvectors. (5 point)

2.3 Probabilistic Pose Classification

Consider the task of pose classification. Let $\theta \in \Theta$ denote the pose class and let I be an image projected into the Eigenspace.

15. Using the Bayes rule, derive an expression for the posterior density of a pose class, given an image $p(\theta|I), k \in \{1 \dots K\}$. (2 point)

To perform probabilistic pose classification, write a program which does the following using the first 50 eigenvectors:

- Finds the Eigenspace representation of the training images of each pose class (use the corresponding Eigenspace based for each pose class)
- Assuming Gaussian distributions, calculates the likelihoods for pose class
- Finds an Eigenspace representation of each testing image for each pose class (use the corresponding Eigenspace)
- Computes the posterior distribution of the pose class for each testing image
- Uses MAP to assign the pose class of the test images
- Calculates the confusion matrix for pose classification

In your report

16. Compare the performance of the probabilistic pose classification with the nearest neighbor classifier, each using the first 50 eigenvectors. (10 point)
17. Display the posterior distribution of the pose class for the same images that you used in Question 14. (5 point)