INT304: Pattern Recognition in Computer Vision **Spring 2024/25**

ASSESSMENT 1: Feature Extraction

Lecturer: Xi Yang *Unit: INT Dept. of SAT*

Disclaimer:

1. *Lab reports deadlines are strict. University's late submission policy will be applied.*
2. *Collusion and plagiarism are absolutely forbidden (University policy will be applied).*
3. *Report due date* 30^{th}*, Mar, 2025*
4. *Contribution to Overall Marks:* 15%
5. *The programming code must be completed in the lab session.*

1.1 Assessment Objectives

This assessment aims at evaluating students' ability to exploit the feature extraction method, which is accumulated during lectures, and after-class study, to analyze, design, implement, develop, test and document the primitive face recognition system using Principle Component Analysis. The assessment will be based on the Pytorch software (Sklearn is allowed to use).

1.2 Dataset (10 points)

In this experiment, we will use the Face dataset to verify our algorithm. The dataset is available at Learning Mall: **INT304-2223-S2 → Sections → Datasets → Face Dataset.**

Split the data into train and test sets using an 80/20 split.

Scaling the faces to a normalized size is a crucial step in facial recognition, and often requires access to a vast amount of training data. Although scikit-learn can handle this task, the primary obstacle is obtaining a sufficient quantity of training data to facilitate successful algorithmic performance.

1) Complete the code and visualize the first 15 images in the dataset. (2')

In the report:

2) It is required to briefly discuss some possible data pre-processing approaches. (5')

3) **Briefly describe the characteristics of the Face dataset. How do you think these characteristics affect the performance of the algorithm? (3')**

1.3 Principal Component Analysis (PCA) and Eigenfaces

A simple facial recognition experiment utilizing PCA and Eigenfaces will be explored. Apply PCA to a face dataset containing images of human faces.

1.3.1 PCA Algorithm (20 points)

PCA can be applied to any matrix. The output of PCA is a set of vectors known as principal components, where each principal component has the length same as the column length of the matrix. It's worth noting that the different principal components from the same matrix are orthogonal to each other.

Standardize the data $\hat{X} = X_{n\times d} - 1_{n\times 1} mean(X)_{1\times d}$

Compute the covariance matrix $\Sigma = \hat{X}^T \hat{X}$

Call the function numpy.linalg.eig(a) to find the most k maximum eigvectors listed as C. Transform the training dataset X using C

$$Y = XC$$

Algorithm 1 PCA Algorithm

1: X: input $n \times d$ data matrix (each row a d-dimensional sample)
2: Standardize the data: subtract mean of X from each row of X
3: Compute the covariance matrix of X (along the row of X) to obtain $\Sigma = (X - \bar{X})^T (X - \bar{X})$
4: Find eigenvectors and eigenvalues of Σ
5: Compute k eigenvectors with largest eigenvalues to construct the matrix $C_{d\times k}$, where the value of eigenvalues gives importance of each component
6: Transform X using C

$$Y = XC$$

where the number of new dimensional is k ($k \ll d$)

In the report:

1) Briefly introduce the basic principles of the PCA algorithm. What is the role of PCA in facial recognition tasks? (12')

2) Discuss the advantages and disadvantages of the PCA algorithm in facial recognition tasks. (8')

1.3.2 Eigenfaces (60 points)

Calculate and visualize Eigenfaces from the given dataset.

Dimensionality reduction will now be performed using PCA by keeping only the first k principal component vectors. If the input matrix is the dataset for face pictures, the first k principal component vectors are the top f most important "face pictures". We call them

the eigenface picture. If we have a weight vector W (column vector), we can add up each eigenfaces (F) subjected to the weight and reconstruct a new face (Z). Then for any W, we can construct the picture of a face as

$$Z = F \cdot W.$$

The following are the specific steps on how to generate Eigenfaces using PCA:

a. Collect dataset: Collect a representative set of facial images. It is recommended that the images are the same size and cropped to the same dimensions.

b. Vectorize: Convert all images to vectors by grouping the grayscale values of each pixel into a vector, and combining all pixel vectors into a large matrix X.

c. Normalize: Center each pixel vector by subtracting the mean of all vectors.

d. Calculate covariance matrix: Calculate the covariance matrix Σ of the centered matrix X.

e. Calculate eigenvalues and eigenvectors: Perform an eigenvalue decomposition of the covariance matrix Σ to obtain the eigenvalues and corresponding eigenvectors. Keep the top k eigenvectors with the largest eigenvalues as the base vectors for generating Eigenfaces.

f. Generate Eigenfaces: Combine the eigenvectors into a new matrix called Eigenfaces. Each row in the matrix corresponds to a specific eigenvector, therefore the number of rows is k and the number of columns is equal to the dimensionality of the pixel vectors.

g. Calculate facial feature vectors: Project each facial image vector onto the generated Eigenface space. The resulting coordinates vector represents the feature vector of that particular face.

h. Face recognition: Compare the distance between two facial feature vectors. The smaller the distance, the more similar the two faces are, and the better the recognition.

1) **Apply PCA to the training data with k components (Allowed to use SKlearn class).** *Code is clear and well-commented (5'). Code is executable and free of errors (5').*

In the report:

2) **Present a detailed experiment setup. (5')**

3) **Visualize EigenFaces for 30 Images. Give an appropriate analysis** (15')**.**

4) **Generate new faces using eigenfaces with random weight vector. Visualize the mean face and random face. It is required to analyze the experimental results.** (15')

5) **Build a face recognition system by using eigenface. Try to tune different factors and give a reasonable analysis.**(15')

1.4 Lab Report (10 points for report quality and format)

1. Write a short report (no more than 7 pages) with clear sections, which should contain a concise description of the methodology, experimental setup, results and observations.
2. Please insert the clipped running image into your report for each step.
3. The report should be formatted using a 12pt Arial font or written with the latex typesetting language. (The Latex Template can be downloaded from module web-page on LearningMall).
4. The report should be pdf format. The naming of report is as follows:
   e.g. INT304Report_StudentID_LastName_FirstName_LabNumber.pdf
   (INT304Report_123456789_Einstein_Albert_1.pdf)

1.5 Hints

Latex IDE: texstudio,overleaf

Python IDE: pycharm

Use the python library: matplotlib, sklearn.

1.6 Marking Scheme:

86%-100% Tasks are well-completed with sufficient and correct analysis. The view of the analysis is properly supported by different aspects. The report is well organized.

70%-85% Essentially complete the tasks with sufficient and correct analysis. The report is carefully organized.

60%-69% Shows understanding but contains a small number of errors or gaps. The report is carefully organized.

40%-59% Clear evidence of a serious attempt at the work, showing some understanding, but with important gaps. Some part of the report is hard to follow.

30%-39% Scrappy work, bare evidence of understanding or significant work omitted. The whole report is very hard to follow.

<30% No understanding or little real attempt made.