VIETNAM NATIONAL UNIVERSITY HOCHIMINH CITY
UNIVERSITY OF SCIENCE
FACULTY OF ELECTRONICS AND TELECOMMUNICATION

END COURSE PROJECT

TOPIC 8

PRINCIPAL COMPONENT ANALYSIS IN IMAGE PROCESSING

Teacher: Ph.D Tran Thi Thao Nguyen

Group Member

Pham Hoai An	21207120
Do Minh Chuong	21207126
Tran Thien Phuc	21207077
Ngo Chon Quang	21207085

Ho Chi Minh City July 21, 2023

Index

Picture list

Table category

1. Introduction

Principal Component Analysis (PCA) is an algorithm that converts a set of data in a multidimensional space to a lower-dimensional space by optimizing the data transformation. The key idea in PCA is to use the eigenvectors of the covariance matrix of the data attributes, arranged in decreasing order of eigenvalues. PCA reduces the dimension of the data by finding the most important principal components in the data while preserving the important information.

This algorithm is commonly used in image processing to reduce computational complexity and decrease the size of image data by finding the principal components of the image. In image processing, PCA can be used to reduce the number of pixels, create a new image based on the principal components, or denoise and reduce the impact of uneven lighting. Additionally, it can be used to create images with evenly distributed colors, clarify lines, improve contrast, and reduce noise.

Overall, choosing a project focused on PCA in image processing can be a great opportunity to explore a powerful and widely used technique, while also contributing to the field of image processing and its practical applications.

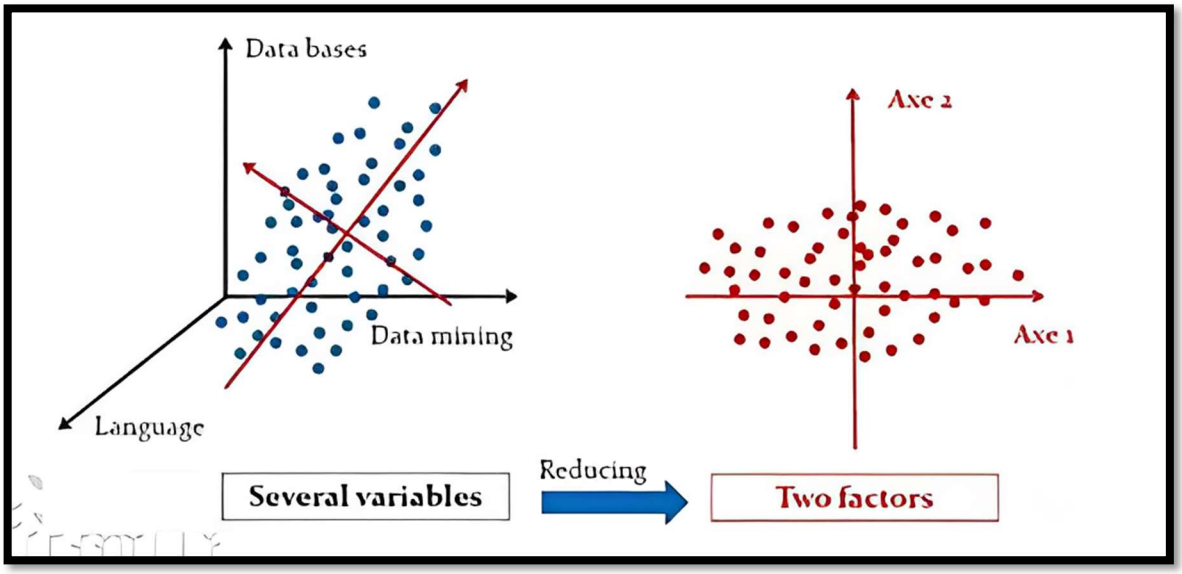

Figure 1. Converting data in multidimensional space to lower dimension space

2. Mathematics

In this section, we attempt to give some necessary mathematical skills to understand the process of Principal Component Analysis.

This section included statistics looks at variance, covariance, and covariance matrix; Linear Algebra looks at eigenvectors and eigenvalues.

2.1 Statistics

2.1.1 Standard Deviation (SD)

Assume that X is our data set:

$$X = [1\ 2\ 4\ 6\ 12\ 15\ 25\ 45\ 68\ 67\ 65\ 98]$$

In this data set, note that X_1 is the first element of data set not X_0 like you may see in some textbooks. Also, the symbol n will be used to refer the number of elements in the X.

The mean of the sample:

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n} \quad (1)$$

If we have two data set as: $[0\ 8\ 12\ 20]$ and $[8\ 9\ 11\ 12]$

The meaning does not tell us a lot about the data except for a sort of middle point. The SD of a data set is a measure of how spread-out data is. The way to calculate it is to compute the squares of the distance from each data point to the mean of the set, add them all up, divide by $n - 1$, and take the positive square root. As formula:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}} \quad (2)$$

2.1.2 Variance

Variance is another measure of the spread of data in a data set. In fact, it is almost identical to the standard deviation. The formula is this:

$$s^2 = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}} \quad (3)$$

You will notice that this is simply the standard deviation squared, in both the symbol (s^2) and the formula (there is no square root in the formula for variance). s^2 is the usual symbol for variance of a sample.

2.1.3 Covariance

Standard deviation and variance only operate on 1 dimension, so that you could only calculate the standard deviation for each dimension of the data set independently of the other dimensions. However, it is useful to have a similar measure to find out how much the dimensions vary from the mean with respect to each other.

Covariance is such a measure. Covariance is always measured between 2 dimensions. If you calculate the covariance between one dimension and itself, you get the variance. So, if you had a 3-dimensional data set (x, y, z), you could measure the covariance between x and y and y or z and z would give you the variance of the x, y and z would give you the variance of the x, y and z dimensions respectively

The formula for covariance is very similar to the formula for variance. The formula for variance could also be written like this:

$$var(X) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})}{(n-1)} \qquad (4)$$

Where I have simply expanded the square term to show both parts. So given that knowledge, here is the formula for covariance:

$$cov(X, Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})}{(n-1)} \qquad (5)$$

2.1.4 Covariance Matrix

Recall that covariance is always measured between 2 dimensions. If we have a data set with more than 2 dimensions, there is more than one covariance measurement that can be calculated. For example, from a 3-dimensional data set (dimensions x, y, z) you could calculate $cov(x, y)$, $cov(x, z)$, and $cov(y, z)$. In fact, for an n-dimensional data set you can calculate $\frac{n!}{(n-2)!*2}$ different convariance values.

A useful way to get all the possible covariance values between all the different dimensions is to calculate them all and put them in a matrix. I assume in this tutorial that you are familiar with matrices, and how they can be defined. So, the definition fir the covariance matrix for a set of data with n dimensions is:

$$C^{m \times n} = \left(c_{i,j}, c_{i,j} = cov\left(Dim_i, Dim_j\right)\right) \qquad (6)$$

2.2 Algebra

This section serves to provide a background for the matrix algebra required in PCA. Specifically, I will be looking at eigenvectors and eigenvalues of a given matrix. Again, I assume a basic knowledge of matrices.

2.2.1 Eigenvalues & Eigenvectors

Eigenvalues and eigenvectors are two important concepts in linear algebra and are widely used in many fields such as mathematics, computer science, and data science. They are related to square matrices, in which the number of columns is equal to the number of rows.

Suppose A is a square matrix of size $n \times n$. If v is a nonzero vector and λ is a scalar, then v is called an eigenvector of A and λ is called the corresponding eigenvalue of A if they satisfy the following equation:

$$Av = \lambda v \quad (7)$$

Here, Av represents the dot product of matrix A and vector v, and λv represents the dot product of scalar λ and vector v.

Another way to understand eigenvectors and eigenvalues is that when a matrix A acts on a vector v to stretch or shrink that vector by a certain scale factor (eigenvalue), then that vector is called an eigenvector corresponding to that eigenvalue. Eigenvectors do not change direction when multiplied by matrix A, but only change their length.

To find eigenvectors and eigenvalues of a matrix, we need to solve the corresponding matrix equation, i.e., find vectors v and scalars λ such that the equation $Av = \lambda v$ holds.

Eigenvalues can be real or complex numbers depending on the properties of the matrix. Eigenvectors corresponding to eigenvalues are often normalized to have a length of 1 for convenience in computation.

Eigenvalues and eigenvectors play an important role in many applications such as spectral analysis, solving systems of linear equations, dimensionality reduction in data science, and many algorithms in computer science and artificial intelligence.

3. PCA

3.1. Definition of PCA

The goal of the PCA technique is to find a lower dimensional space or PCA space (W) that is used to transform the data $X = \{x_1, x_2, \dots, x_n\}$ from a higher dimensional space (R^M) to a lower dimensional space R^k, where N represents the total number of samples or observations and x_i represents t^{th} sample, pattern, or observation. All samples have the same dimension $(x_i \in R^m)$. In other words, each sample is represented by M variables. The direction of the PCA space represents the direction of the maximum variance of the given data as shown in Figure 1. As shown in the figure, the PCA space consists of a number of PCs. Each principal component has a different robustness according to the amount of variance in its direction.

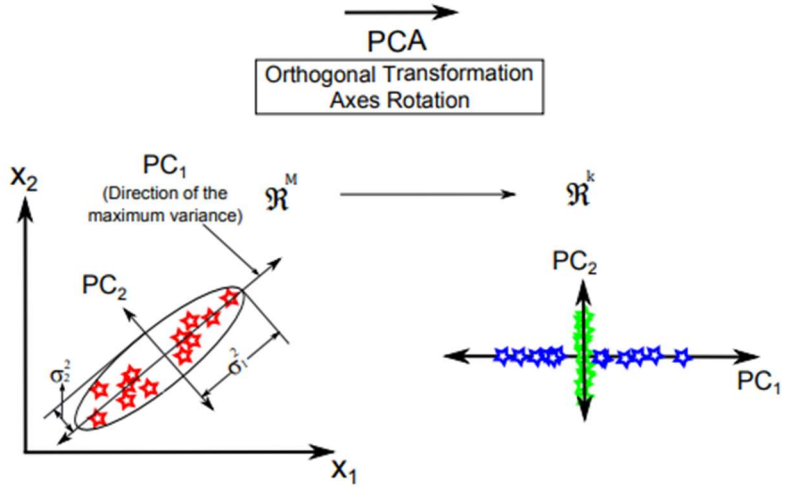

Figure 2. Example of the two-dimensional data (x1, x2). The original data are on the left with the original coordinate, i.e., x1 and x2, the variance of each variable is graphically represented and the direction of the maximum variance, i.e., the principal component P C1, is shown; on the right the original data are projected on the first (blue stars) and second (green stars) principal components.

3.2. Principal Components (PCs)

The PCA space consists of k principal components. The principal components are orthonormal[a], uncorreclated[b], and it represents the direction of the maximum variance.

The first principal component $((PC_1 \; or \; v_1) \in R^{M \times 1})$ of the PCA space represents the direction of the maximum variance of the data, the second principal component has the second largest variance, and so on. Figure 1 shows how the original data are transformed from the original space (R^M) to the PCA space (R^k). Thus, the

PCA techique is considered an orthogonal transformation due to its orthogonal principal components or axes rotation due to the rotation of the original axes (Wold et al., 1987; Shlens, 2014). There are two methods to calculate the principal components. The first method depends on calculating the covariance matrix, while the second one uses the SVD method.

3.3. Covariance Matrix Method

In this method, there are two main steps to calculate PCs of the PCA space. First, the covariance matrix of the data matrix (X) is calculated. Second, the eigenvalues and eigenvectors of the covariance matrix are calculated. Figure 3 illustrates the visualized steps of calculating the PCs using the covariance matrix method.

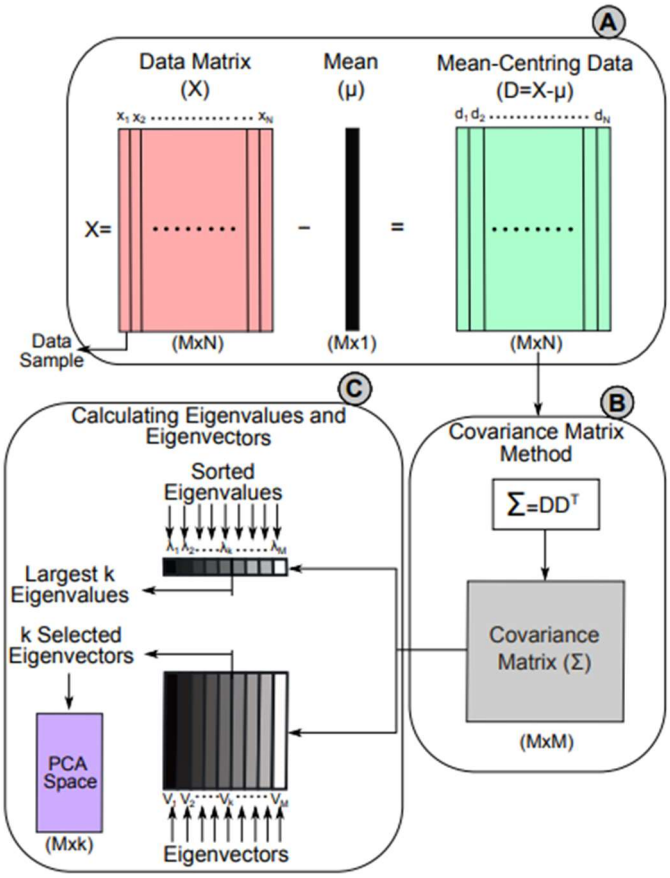

Figure 3. Visualized steps to calculate the PCA space using the covariance matrix method.

3.3.1. Calculate the Covariance Matrix (∑):

The variance of any variable measures the deviation of that variable from its mean value and it is defined as follows, $\sigma^2(x) = Var(x) = E((x-\mu)^2) = E\{x^2\} - (E\{X\})^2$, where μ represents the mean of the variable x, and $E(x)$ represents the expected value of x. The covariance matrix is used when rhe number of variables

more than one and it is defined as follows, $\sum_{ij} = E\{x_i y_j\} - E\{x_i\}E\{x_j\} = E\left[(x_i - \mu_i)(x_j - \mu_j)\right]$. As shown in Figure 3, step(A), after calculating the mean of each variable in the data matrix, the mean-centring data are calculated by suntracting the mean ($\mu \in R^{M \times 1}$) from each sample as follows, $D = \{d_1, d_2, \ldots, d_N\} = \{x_1 - \mu, x_2 - \mu, x_N - \mu\}$. The covariance matrix is then calculated as follows, $\sum = DD^T$(Figure 3, step (B)).

Covariance matrix is a symmetric matrix (i.e., $X = X^T$) and always positive semi-definite matri. The diagonal values of the covariance matrix represent the variance of the variable $x_i, i = 1, \ldots, M$, while the off-diagonal entries represent the covariance between two different variables as shown in Equation (1). A positive value in covariance matrix means a positive correlation between the two variables, while the negative value indicates a negative correlation and zero value indicates that the two variables are uncorrelated or statistically independent.

$$\begin{matrix} var(x_1, x_1) & var(x_1, x_2) & \ldots & cov(x_1, x_M) \\ cov(x_2, x_1) & cov(x_2, x_2) & \cdots & cov(x_2, x_M) \\ \vdots & \vdots & & \vdots \\ cov(x_M, x_1) & cov(x_M, x_2) & \ddots & var(x_M, x_M) \end{matrix}$$

3.3.2. Calculate Eigenvalue (λ) and Eigenvector (V).

The covariance matrix is solved by calculating the eigenvalues (λ) and eigenvectors (V) as follows:

$$V\Sigma = \lambda V \quad (8)$$

where V and λ represent the eigenvectors and eigenvalues of the covariance matrix, respectively.

The eigenvalues are scalar values, while the eigenvectors are non-zero vectors, which represent the principal components, i.e., each eigenvector represents one principal component. The eigenvectors represent the directions of the PCA space, and the corresponding eigenvalues represent the scaling factor, length, magnitude, or the robustness of the eigenvectors. The eigenvector with the highest eigenvalue represents the first principal component and it has the maximum variance as shown in Figure 2. The eigenvalues may be equal when the PCs have equal variances and hence all the eigenvectors are the same and we cannot decide which eigenvectors are used to construct the PCA space.

3.4. PCA space (Lower Dimensional Space)

To construct the lower dimensional space of PCA (W), a linear combination of k selected PCs that have the most k eigenvalues are used to preserve the maximum amount of variance, i.e., preserve the original data, while the other eigenvectors or PCs are neglected as shown in Figures 3(step C) and 3(step C). The lower dimensional space is denoted by $W = \{v1, \ldots, vk\}$. The dimension of the original data is reduced by projecting it after subtracting the mean onto the PCA space as in Equation (9):

$$Y = W^T D = \sum_{i=1}^{T} W^T (x_i - \mu) \quad (9)$$

Where $Y \in R^k$ represents the original data after projecting it onto the PCA space as shown in Figure 4, thus $(M - k)$ features or variables are lost from the original data

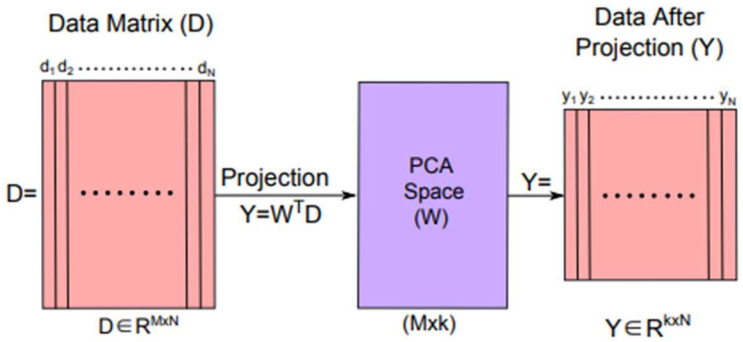

Figure 4. Data projection in PCA

3.5. Data Reconstruction

The original data can be reconstructed again as in Equation (10).

$$\hat{X} = WY + \mu = \sum_{i=1}^{N} W y_i + \mu \quad (10)$$

where $\hat{X}$ represents the reconstructed data. The deviation between the original data and the reconstructed data is called the reconstruction error or residuals as denoted in Equation (11). The reconstruction error represents the square distance between the original data and the reconstructed data, and it is inversely proportional to the total variance of the PCA space. In other words, selecting a large number of PCs increases the total variance of W and decreases the error between the reconstructed and the original data. Hence, the robustness of the PCA is controlled by the number

of selected eigenvectors (k) and it is measured by the sum of the selected eigenvalues, which is called total variance as in Equation (12). For example, the robustness of the lower dimensional space $W = \{v_1 \ldots, v_k\}$ is measured bythe ratio between the total variance $(\lambda_i, i = 1, \ldots, k)$ of W to the total variance.

$$Error = X - \hat{X} = \sum_{i=1}^{N} (x_i - \hat{x_i})^2 \quad (11)$$

$$\text{Robustness of the PCA space} = \frac{\text{Total Variance of W}}{\text{Total Variance}} = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{M} \lambda_i} \quad (12)$$

3.6. PCA Algorithms

Algorithms 1: Calculate the PCs using Covariance Matrix Method

1: Given a data matrix $X = [x_1, x_2, \ldots x_N]$, where N represents the total number of samples and x_i represents the i^{th} sample.

2: Compute the mean of all sample

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i \quad (13)$$

3: Subtract the mean from all samples:

$$D = \{d_1, d_2, d_N\} = \sum_{i=1}^{N} x_i - \mu \quad (14)$$

4: Compute the Covariance Matrix:

$$\sum = \frac{1}{N-1} D \times D^T \quad (15)$$

5: Compute the eigenvectors V and eigenvalues λ of the covariance matrix (Σ).

6: Sort eigenvectors according to their corresponding eigenvalues.

7: Select the eigenvectors that have the largest eigenvalues $W = \{v_1, \ldots, v_k\}$. The selected eigenvectors (W) represent the projection space of PCA.

8: All samples are projected on the lower dimensional space of PCA (W) as follows, $Y = W^T D$.

3.7. Algorithm in Python and Source code

3.7.1. There are sever steps to perform PCA in image processing:

Step 1: Data preparation: Construct a data matrix, where each row is a sample, and each column is a variable.

Step 2: Subtract the mean value: Before performing PCA, it is usually necessary to subtract the mean value of each variable to ensure that the data is centered at the origin.

Mean value of each variable:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} X_i$$

Subtract mean value out of matrix X:

$$X_{centered} = X - \mu$$

Step 3: Calculate the covariance matrix: Calculate the covariance matrix of the data, which represents the correlation between variables.

Find covariance matrix:

$$C = \frac{1}{N} X_{centered}^{T} X_{centered}$$

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix: This is an important step in PCA, as it determines the principal components and their importance.

Find vectors and own value of covariance matrix:

$$C_{v_i} = \lambda_i v_i$$

Step 5: Select the principal components: Sort the eigenvalues in descending order and select a number of principal components such that the total selected eigenvalues achieve sufficient data variance (determined beforehand by a threshold or percentage of retained variance).

Step 6: Project the data onto the principal components: Calculate the projection values of the data onto the selected principal components to obtain a compact and reduced representation of the data.

$$Y = X_{centered} \,.\, V$$

Step 7: Optional: If desired, perform inverse PCA to transform the compact data representation back into the original data space.

$$X_{approx} = Y \,.\, V^t + mean$$

3.7.2.Python Source Code for PCA in Image Processing

```
import cv2
import numpy as np
import matplotlib.pyplot as plt


def pca(image, num_components):
    image_height, image_width, num_channels = image.shape
    X = image.reshape(-1, num_channels)

    X_centered = X - np.mean(X, axis=0)


    C = np.cov(X_centered, rowvar=False)

    eigenvalues, eigenvectors = np.linalg.eig(C)

    sorted_indices = np.argsort(eigenvalues)[::-1]
    sorted_eigenvalues = eigenvalues[sorted_indices]
    sorted_eigenvectors = eigenvectors[:, sorted_indices]

    principal_components = np.dot(X_centered, sorted_eigenvectors[:, :num_components])

    reconstructed_image = np.dot(
        principal_components, sorted_eigenvectors[:, :num_components].T
    )
    reconstructed_image = reconstructed_image + np.mean(X, axis=0)
    reconstructed_image = np.uint8(
        reconstructed_image.reshape(image_height, image_width, num_channels)
    )

    return principal_components, reconstructed_image
```

```
image_path = "OIP.jpeg"
image = cv2.imread(image_path)

num_components = 2

principal_components, reconstructed_image = pca(image, num_components)

print("Các thành phần chính:")
print(principal_components)

plt.subplot(1, 2, 1)
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
plt.title("Hình ảnh gốc")
plt.axis("off")

plt.subplot(1, 2, 2)
plt.imshow(cv2.cvtColor(reconstructed_image, cv2.COLOR_BGR2RGB))
plt.title("Hình ảnh tái tạo từ PCA")
plt.axis("off")

plt.show()
```

Figure 5. Result of PCA in image processing

4. Application

In digital signal processing, PCA is usually used to handle large data sizes that are difficult to process and understand. It allows for dimensionality reduction by selecting the most important principal components while minimizing data loss.

Denoising: In many cases, digital signals may be disturbed and affect noise. Using PCA can eliminate some of the noise and produce a cleaner signal.

Signal analysis: PCA is used to correctly understand the structure and the relationships of digital signal components.

5. Advantage & disadvantage

Advantage:

PCA is a useful tool in cutting off dimension of data, especially in this case has multidimensional input and relationship. When data dimension is lower, it is easy to create figure and analysis it simultaneously reduced calculated cost.

Disadvantage:

PCA is only used for continuous variables and not suitable for dividing variables. Besides, lower dimensions may lose many important original data.

6. Conclusion

PCA is strongly tools in lowing dimensions of data and finding principal components. However, as with any tools, it has a lot of restrictions that need to be considered to apply in real life.

7. References

[1] “Principal components analysis”, Available: https://en.wikipedia.org/wiki/Principal_component_analysis, Visit on: 7/25/2023

[2] “Principal Component Analysis (phần 1/2)”, Available: https://machinelearningcoban.com/2017/06/15/pca/, Visit on: 7/25/2023

[3] “Principal Component Analysis (phần 2/2)”, Available: https://machinelearningcoban.com/2017/06/21/pca2/, Visit on: 7/25/2023

[4] Johnathan Shlens, “A Tutorial on Principal Components Analysis”, April 7, 2014

[5] Lindsay I Smith, “A tutorial on Principal Components Analysis”, February 26, 2002

[6] Alaa Tharwat, Principal Component Analysis – A Tutorial, Article in International Journal of Applied Pattern Recognition · January 2016

Table 1. Participants rating

Members	Student ID	Tasks	Rating	Note
Pham Hoai An	21207120	Simulation/Demo Code	95	
Tran Thien Phuc	21207077	Making PowerPoint	95	
Do Minh Chuong	21207126	Making PowerPoint	95	Leader
Ngo Chon Quang	21207085	Writing report	95	

Score scale (%)