Face Recognition - Eigenface Technique

Ugenteraan Manogaran

March 2019

Contents

1.0 Dataset

Suppose there are m number of n x n grayscale face images **(Strictly face images. Positioning of each face must be consistent between images.)**.

- Each of the images can be represented as an n x n matrix

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & . & . & . & x_{1n} \\ x_{21} & x_{22} & . & . & . & x_{2n} \\ . & . & . & & & . \\ . & . & & . & & . \\ . & . & & & . & . \\ x_{n1} & x_{n2} & . & . & . & x_{nn} \end{bmatrix}$$

  or, equivalently each of them can be represented as an n^2-vector where each of the columns are stacked below the previous columns as such :

$$\mathbf{x} = \begin{bmatrix} x_{11} \\ . \\ . \\ . \\ x_{n1} \\ x_{12} \\ . \\ . \\ . \\ x_{nn} \end{bmatrix}$$

- For convenience purpose, we will rewrite $\mathbf{x}$ as

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_{n^2} \end{bmatrix}$$

- Now, all the m images can be represented as an n^2 x m matrix $\mathbf{A}$ where the superscript (i) will be used to denote the *i-th* image. Hence, i $\in 1, \ . \ . \ . \ , m$.

$$\mathbf{A} = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & . & . & . & x_1^{(m)} \\ x_2^{(1)} & x_2^{(2)} & . & . & . & x_2^{(m)} \\ . & . & . & & & . \\ . & . & & . & & . \\ . & . & & & . & . \\ x_{n^2}^{(1)} & x_{n^2}^{(2)} & . & . & . & x_{n^2}^{(m)} \end{bmatrix}$$

2.0 Problem Statement

It is entirely possible to perform *facial recognition* [1] by comparing the distance between the given face image $\hat{\boldsymbol{x}}$, and each image $\mathbf{x}^{(i)}$, from the dataset.

$$d_i \; = \| \; \hat{\boldsymbol{x}} - \mathbf{x}^{(i)} \; \| \tag{1}$$

Whenever the value of a d_i is smaller than a chosen threshold α, the given face image can be identified as the person that is associated with the *i-th* face image.

However, the vectors that represent the face images often have large dimensionality (n^2-dimensional vectors). Even if the face images are relatively small in size such as 256 by 256 pixels, the vectors would have a dimension of 65,536 or equivalently, a point in 65,536-dimensional space. Performing computations on this huge dimensional vectors would be time consuming and computationally expensive. Hence, the need to perform facial recognition using other methods are needed.

Since the images of faces are similar in overall configuration, the vectors that represent the images will not be randomly distributed in this huge *image space*[2]. Therefore the image space can be described by a lower-dimensional subspace. How do we find this lower-dimensional subspace?

3.0 Eigenfaces

3.1 Principal Component Analysis

Principal Component Analysis (PCA) is an algorithm that reduces the dimensionality of a data set to a lower-dimensional linear subspace by linear projection in such a way that the reconstruction error made by the linear projection is as low as possible.

NOTE : *To know more about PCA, visit* : *Principal Component Analysis*

3.2 PCA and Eigenfaces

Using PCA, we can find the vectors that best account for the distribution of face images within the entire image space. In other words, PCA would be able to capture the variation in a collection of face images and use this information to encode the face images.

Therefore, we would like to find the eigenvectors of the covariance matrix of the set of face images. Since each of the vectors that represent the face images contributes more or less to each eigenvector, each eigenvector, when viewed as an image, would look like some sort of ghostly face that we call eigenface. Each individual face can be represented exactly in terms of a linear combination of

[1] Problem of identifying a specific individual by matching the given face image with previously stored face images.

[2] The n^2-dimensional space.

the eigenfaces.

Furthermore, each of the face images can be approximated using only the eigenfaces/eigenvectors that are associated with the largest eigenvalues. The best q eigenfaces that spans a q-dimensional subspace is known as face space.

3.3 Calculating Eigenfaces

As mentioned in **section 1.0**, $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ . \ . \ . \ , \mathbf{x}^{(m)}$ is the set of face images. The mean face image $\boldsymbol{\mu}$ of the set is defined by

$$\boldsymbol{\mu} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}^{(i)} \tag{2}$$

Each of the face images differ from the mean by the vector $\widetilde{\boldsymbol{x}}$ defined by

$$\widetilde{\boldsymbol{x}}^{(i)} = \mathbf{x}^{(i)} - \boldsymbol{\mu} \tag{3}$$

All of the $\widetilde{\boldsymbol{x}}^{(i)}$ can be represented in a n^2 x m matrix $\widetilde{\boldsymbol{A}}$ as

$$\widetilde{\boldsymbol{A}} = [\widetilde{\boldsymbol{x}}^{(1)} \quad \widetilde{\boldsymbol{x}}^{(2)} \quad . \quad . \quad . \quad \widetilde{\boldsymbol{x}}^{(m)}]$$

NOTE : *Each of the* $\widetilde{\boldsymbol{x}}^{(i)}$ *are vectors.*

The n^2 x n^2 covariance matrix of $\widetilde{\boldsymbol{A}}$, $\mathbf{C}$ is defined by

$$\mathbf{C} = \frac{1}{m}\sum_{i=1}^{m}(\widetilde{\boldsymbol{x}}^{(i)}\widetilde{\boldsymbol{x}}^{(i)\mathbf{T}}) \qquad or \qquad \mathbf{C} = \frac{1}{m}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{A}}^{\mathbf{T}} \tag{4}$$

Recall that each of the face images can be represented as a linear combination of all the eigenvectors of $\mathbf{C}$. However, calculating the eigenvectors of $\mathbf{C}$ is a computationally expensive task since C is a relatively huge matrix. Fortunately, it can be shown that the m x m symmetric matrix $\frac{1}{m}\widetilde{\boldsymbol{A}}^{\mathbf{T}}\widetilde{\boldsymbol{A}}$ has m eigenvalues that are also eigenvalues of $\mathbf{C}$.

$$\frac{1}{m}\widetilde{\boldsymbol{A}}^{\mathbf{T}}\widetilde{\boldsymbol{A}}\mathbf{v}_i \ = \ \lambda_i\mathbf{v}_i \implies \widetilde{\boldsymbol{A}}(\frac{1}{m}\widetilde{\boldsymbol{A}}^{\mathbf{T}}\widetilde{\boldsymbol{A}}\mathbf{v}_i) \ = \ \widetilde{\boldsymbol{A}}(\lambda_i\mathbf{v}_i) \implies \frac{1}{m}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{A}}^{\mathbf{T}}\widetilde{\boldsymbol{A}}\mathbf{v}_i \ = \ \lambda_i\widetilde{\boldsymbol{A}}\mathbf{v}_i$$
$$\implies \mathbf{C}\widetilde{\boldsymbol{A}}\mathbf{v}_i \ = \ \lambda_i\widetilde{\boldsymbol{A}}\mathbf{v}_i$$

From above, it can be concluded that both $\frac{1}{m}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{A}}^{\mathbf{T}}$ and $\frac{1}{m}\widetilde{\boldsymbol{A}}^{\mathbf{T}}\widetilde{\boldsymbol{A}}$ have the same eigenvalues. This implies that, if $m < n^2$, then there will only be m, rather than n^2 *meaningful*[3] eigenvectors. All the remaining eigenvectors will have associated eigenvalues of zero.

[3]Eigenvectors that are associated with nonzero eigenvalues.

Therefore, we construct the m x m symmetric matrix $\mathbf{L} = \frac{1}{m}\widetilde{\boldsymbol{A}}^{\mathbf{T}}\widetilde{\boldsymbol{A}}$ and find its eigenvectors $\mathbf{v}_i$. Using the eigenvectors of $\mathbf{L}$, the eigenvectors of $\mathbf{C}$ can be found by $\widetilde{\boldsymbol{A}}\mathbf{v}_i$. With this method, the computational power needed to find the eigenvectors of $\mathbf{C}$ is greatly reduced. The associated eigenvalues can be used to rank the eigenvectors in a descending order.

NOTE : The eigenvectors $\widetilde{\boldsymbol{A}}\mathbf{v}_i$ *are n^2-dimensional vectors.*

4.0 Face Detection and Recognition

The m eigenvectors calculated from $\mathbf{L}$ spans a basis set that describes the face images. However, in practice, a smaller number of eigenvectors is sufficient for identification. Therefore, we will select the best q eigenvectors such that $q < m$ based on our ranking. These q eigenvectors are our eigenfaces and the space spanned by them is our face space.

A new image $\hat{\boldsymbol{x}}$ (n^2-vector) is first subtracted by the mean vector from the set of images $\widetilde{\boldsymbol{x}}$. The vector $\hat{\boldsymbol{x}} - \boldsymbol{\mu}$ is then projected to the face space and represented by

$$\hat{\boldsymbol{x}}_{proj} = \sum_{i=1}^{q}(\mathbf{u}_i^{\mathbf{T}}(\hat{\boldsymbol{x}} - \boldsymbol{\mu}))\mathbf{u}_i \tag{5}$$

where

$$\mathbf{u}_i = \frac{\widetilde{\boldsymbol{A}}\mathbf{v}_i}{\| \widetilde{\boldsymbol{A}}\mathbf{v}_i \|}$$

The vector $\hat{\boldsymbol{x}}_{proj}$ is the new face image represented by the linear combination of the eigenfaces. The more the eigenfaces used for the reconstruction, the more the reconstructed face image looks like the original new face image. Notice that $\hat{\boldsymbol{x}}_{proj}$ and the eigenfaces are n^2-vectors. The vector $\hat{\boldsymbol{x}} - \boldsymbol{\mu}$ can also be represented as a q-vector with respect to the eigenfaces, $\mathbf{\Omega}$

$$\mathbf{\Omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ . \\ . \\ . \\ \omega_q \end{bmatrix} \tag{6}$$

where

$$\omega_i = \mathbf{u}_i^{\mathbf{T}}(\hat{\boldsymbol{x}} - \boldsymbol{\mu}) \quad \text{for } i \in 1,\ 2,\ \ldots, q \tag{7}$$

4.1 Face Detection

To determine whteher the given image $\hat{\boldsymbol{x}}$ is a face image, the distance of $\hat{\boldsymbol{x}}$ from the face space can be used. The distance of $\hat{\boldsymbol{x}}$ from the face space β is

$$\beta \;=\; \sqrt{\| \;\hat{\boldsymbol{x}} \;-\; \hat{\boldsymbol{x}}_{proj} \,\|} \tag{8}$$

If β is smaller than a chosen threshold α_1, then $\hat{\boldsymbol{x}}$ can be categorized as a face image.

4.2 Face Recognition

To determine which face class provides the best description of the new image, we need to find the face class k that minimizes the Euclidean distance

$$\epsilon_k \;=\; \sqrt{\| \;(\boldsymbol{\Omega} \;-\; \boldsymbol{\Omega}^{(k)}) \,\|} \quad \text{for } k \in 1,\; 2,\; . \; . \; . \;, m \tag{9}$$

where $\boldsymbol{\Omega}^{(k)}$ is the vector of $\widetilde{\boldsymbol{x}}^{(k)}$ represented with respect to the eigenfaces. Therefore,

$$\boldsymbol{\Omega}^{(k)} \;=\; \begin{bmatrix} \omega_1^{(k)} \\ \omega_2^{(k)} \\ . \\ . \\ . \\ \omega_q^{(k)} \end{bmatrix} \tag{10}$$

where

$$\omega_i^{(k)} \;=\; \mathbf{u}_i^{\mathbf{T}} \widetilde{\boldsymbol{x}}^{(k)} \quad \text{for } i \;\in\; 1,\; 2,\; . \; . \; . \;, q \tag{11}$$

If the smallest ϵ_k is lower than a chosen threshold α_2, the new face image $\hat{\boldsymbol{x}}$ is classified as face k.

References

[1] *Turk, M. A., and Pentland, A. P. (1991, June). Face recognition using eigenfaces. In Proceedings 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 586-591). IEEE.*

[2] *Huang, K. (2012). Principal Component Analysis in the Eigenface Technique for Facial Recognition.*