Face Recognition Using PCA , 2d-PCA and k-PCA

*Mathematics For Intelligent Systems -3

1st Dhanvanth Reddy Yerramreddy
dept. of Cse(AI)
Amrita Vishwa Vidyapeetham
Kollam, Kerala
AM.EN.U4AIE20077

2nd Paladugula Pruthvi
dept. of Cse(AI)
Amrita Vishwa Vidyapeetham
Kollam, Kerala
AM.EN.U4AIE20054

3rd Nehith Sai Vemulapalli
dept. of Cse(AI)
Amrita Vishwa Vidyapeetham
Kollam, Kerala
AM.EN.U4AIE20053

4th Aryan kamulagari
dept. of Cse(AI))
Amrita Vishwa Vidyapeetham
Kollam, Kerala
AM.EN.U4AIE20014

***Abstract*—It is commonly understood that effective biometric identification is becoming increasingly important. In the subject of biometrics, human face recognition is a crucial component. It has been a focus of research for decades, but due to the complexity of the human face, it remains a difficult challenge to solve.**

Face recognition requires good computational algorithms because it is a complicated multidimensional structure. Face recognition is treated as a two-dimensional problem in our approach. Face recognition is accomplished using Principal Component Analysis in this technique (PCA). Face photos are projected into a face space that encapsulates the most diversity among known face images. The face space is described by eigenface which are eigenvectors of the set of faces, which may not match general facial features such as eyes, nose, mouth. The PCA is used in the eigenface technique to recognize images. The method works by projecting a pre-extracted face image onto a set of face space that represents substantial differences between known face images. After comparing with the current database, the faces will be classified as known or unknown. If the user is new to the face recognition system, his or her template will be saved in the database, otherwise, it will be matched against the database's templates. The narrower face space than the training set of faces is explained by PCA's variable lowering hypothesis.

I. Introduction

Automatic face recognition has gotten a lot of interest in recent years, and its research has gotten a lot of attention from not only engineers but also neuroscientists because it has a lot of potential applications in computer visual communication and automatic access control systems. A computer application capable of detecting or validating a person from a digital image or a video frame from a video source is known as a face recognition system. One of the methods to do this is by comparing chosen face traits from the image and a facial database. It's commonly utilized in security systems, and it's similar to other biometrics like fingerprint or eye iris recognition systems. Recently, it has also become popular as a commercial identification and marketing tool.

Face recognition algorithms have been proposed numerous times over the previous few decades. All of these methods for recognizing human faces can be split into two categories: geometrical features (In this technique, a set of geometrical features such as nose width and length, mouth position, and chin shape are computed.) This set of features is then compared to the features of known individuals using the Euclidean distance and template matching (here we are not trying to classify an image as a "face" or "non-face," but rather trying to recognize a face by extracting the entire facial regions: matrix of pixels, and comparing these to the stored images of known individuals). Whereas a feature-based strategy may provide faster recognition and need less memory, template-based techniques provide improved recognition accuracy).

Many features are not effective for obtaining a desirable learning result in many patterns sets with a big number of features and a small number of observations, such as bioinformatics data, and the short observations may cause the learning algorithm to overfit to the noise. Then, by selecting a better feature space, you can solve a variety of Pattern Recognition difficulties. The suitable features should be resistant against geometry distortion induced by changing viewpoints, as well as informative to capture an object's uniqueness. Using such attributes has the advantages of a shorter inverted list, which saves memory, more relevance for the result candidate set, and faster recognition through expedited ranking.

Principal Components Analysis (PCA) is a method for transforming a dataset's columns into a new set of features known as Principal Components. By doing so, a big portion of the data from the entire dataset is effectively compressed into a smaller number of feature columns. This allows for dimensionality reduction and the visualization of any class or cluster separation. In general, the PCA is a method for dimensionality reduction that is largely used for data analysis.

In a nutshell, PCA allows you to decrease the features of a dataset while still retaining the majority of the data.

The analysis is simplified in terms of processing costs while maintaining a high level of accuracy. As a result, this method is beneficial in face recognition when the data being analysed is photos with a significant number of variables (pixels).

The PCA is an orthogonal linear transformation that transfers data to a new coordinate system, with the first new components accounting for the majority of the variance. The eigenvectors of the covariance matrix of the input data make up the change of base matrix utilised to construct this projection. These eigen vectors are also known as eigenfaces in the context of face recognition.

The conventional PCA's goal is the same as the 2DPCA's. The PCA, on the other hand, requires input data in the form of a matrix, but the 2DPCA can be applied straight to a tensor. This method has found to be useful in some situations, such as colour face recognition. In this situation, the eigen decomposition approach is applied instead of SVD. This procedure entails determining the tensor's covariance matrix and calculating its eigen vectors in decreasing order of the corresponding eigenvalues.

Kernel Principal Component Analysis (KPCA) is a dimensionality reduction method that is nonlinear. It uses kernel approaches to extend Principal Component Analysis (PCA), which is a linear dimensionality reduction methodology. Sometimes the structure of the data is nonlinear, and the principle components will not provide us the ideal dimensionality reduction, thus we employ non-linear approaches like KPCA . The concept behind KPCA is that many datasets that aren't linearly separable in their original space can be made linearly separable by projecting them onto a higher dimensional environment.On the initial data dimensions, the extra dimensions are merely basic arithmetic operations.

Steps of KPCA

- We'll start by choosing a kernel function k(xi, xj) and a transformation to a higher dimension, T.
- We'll also find the covariance matrix of our data, similar to PCA. However, to calculate this matrix, we shall use the kernel function. As a consequence, the kernel matrix, which is the matrix obtained by applying the kernel function to all pairs of data, will be computed.
- Center our kerenl matrix (this corresponds to substract the mean of the converted data and dividing by standard deviations)
- Then, we shall find eigenvectors and eigenvalues of this matrix.
- Sort our eigenvectors in decreasing order depending on their corresponding eigenvalues.
- We'll decide on the amount of dimensions our reduced dataset should have, let's name it m. Then we'll concatenate our first m eigenvectors into a single matrix.
- Finally, multiply your data by the product of that matrix. As a result, you'll have a new, smaller dataset.

A. *BioMetrics*

Biometrics is used to verify or identify that a person seeking a network resource is who he, she, or it claims to be, and vice versa. It makes use of the property that a human trait is connected with a person, such as finger structure, face details, and so on. We can confirm a person's identification by comparing the existing data with the incoming data. Fingerprint recognition, face detection and recognition, iris recognition, and other biometric systems are used for human identification in surveillance systems and criminal identification. The use of these qualities for identification has the advantage of not being forgotten or lost. These are distinct characteristics of a human being that are extensively used.

B. *Face Recognition*

The Face Recognition system is made up of a series of tasks. We used these procedures steps in our study to solve the facial recognition challenge using the face94 and face95 databases.

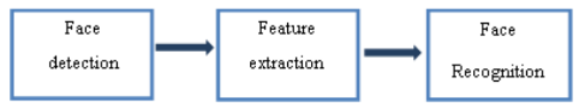


Fig. 1. The Face Recognition System

Face recognition requires good computational algorithms because it is a complicated multidimensional structure. In social situations, the face is our primary and first centre of attention, and it plays a significant part in determining an individual's identity. Even after years, we can recognise a number of faces we've learnt over our lives and identify them at a look. Faces may alter as a result of ageing and distractions such as beards, spectacles, or haircut changes.

Biometrics includes face recognition as a key component. Basic human qualities are matched to existing data in biometrics, and a human being's identity is traced based on the results of the matching. Facial traits are extracted and implemented using efficient algorithms, with certain changes made to improve the existing method models.

Face-detection and recognition computers could be used in a range of practical applications, such as criminal identification, security systems, and identity verification. Face detection and identification are currently employed in a variety of venues, including image storage services and social networking sites. Face recognition and detection can be accomplished utilising computer science methods.

Face features are extracted and compared to faces in the database that have been similarly processed. If a face is recognised, it is known; otherwise, the system may display a similar face already in the database. If an unfamiliar face shows more than once in a surveillance system, it is saved in a database for further recognition. These steps are quite helpful in identifying criminals. Face Features are retrieved from a face and processed before being compared to similarly processed faces in the database. If a face is recognised, it is known; otherwise, the system may display a similar face already in the database.recognition techniques can be divided into two groups based on the face representation they use appearance-based, which uses holistic texture features and is applied to either whole face or specific regions in a face image and feature-based, which uses geometric facial features (mouth, eyes, brows, cheeks etc), and geometric relationships between them.

C. Face Detection

This is one of the most important processes, which is described as the process of determining and extracting faces from input photos or video images. Segmentation, extraction, and verification of faces and maybe facial features from an uncontrolled background are some of the methods that may be utilised to complete this task.Features Extraction

D. Features Extraction

The process of extracting facial features from input photos is classified as this task. Face areas, variations, angles, or metrics might be used, and they could be human-relevant (e.g. eye spacing) or not. Other applications of this phase include face feature tracking and emotion recognition. We are focused in this process on PCA, 2d PCA, KPCA of extracting important information from the face, as we evaluated methods, as shown in Figure , and as follows :

- Apply directly on pixel image intensities
- Apply on DCT Coefficients of image ,
- And apply on FFT Coefficients of image

After taking the overall averages, the above tests were applied to the training data, and they were also applied in the case of taking the average of each class data.

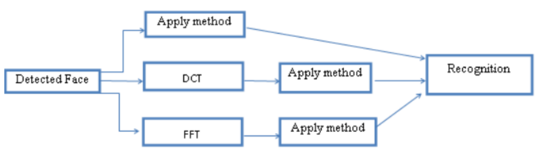


Fig. 2. The General Outline of Feature Extraction

II. Literature survey

A. Principal Component Analysis (PCA)

Karl Pearson invented principal component analysis (PCA) in 1901. PCA is a variable reduction approach that is useful when there is some duplication in the data. As a result, variables will be reduced to a smaller set of variables known as Principal Components, which will account for the majority of the variance in the observed variable. When we want to do recognition in a high-dimensional space, we run into issues. The goal of PCA is to keep as much variety in our original data set as feasible while reducing the dimensionality of the data. Dimensionality reduction, on the other hand, entails information loss. The best principal components can be used to find the optimum low-dimensional space.

Analyze the principal components PCA is a typical technique for data reduction and feature extraction in statistical pattern recognition and signal processing. It's a method for identifying patterns in data and expressing it so that similarities and differences can be discovered. Analyzing patterns in highdimensional data can be challenging, but PCA is an useful tool for dealing with such data. Another advantage of PCA is that it prevents the loss of vital information by limiting the number of dimensions used to discover patterns. The PCA face recognitionalgorithm finds eigenvectors, also known as eigenfaces, which represent the training images' global feature. Y1, Y2,..., YN:

- Compute the mean vector
  a. $Me = \frac{1}{N}\sum_{i=1}^{N} Y_i$ (4)
- Compute the *covariance matrix*
  b. $C = \frac{1}{N}\sum_{i=1}^{N}(Y_i - Me)(Y_i - Me)^t$ (5)
- Compute the eigenvalue/eigenvector pairs (λi, ui) of C, $1 \le i \le N$, where $\lambda 1 \ge \lambda 2 \ge \cdots \ge \lambda_N$.
- Compute the first k principal components $z_j^{(i)} = Yj^t ui$, for each observation Y_j, $1 \le j \le n$, along the direction ui, $i = 1, 2, \cdots, k$. where $z_j^{(i)}$ is the eigenface.

The conventional PCA's goal is the same as the 2DPCA's. The PCA, on the other hand, requires input data in the form of a matrix, but the 2DPCA can be applied straight to a tensor. This method has found to be useful in some situations, such as colour face recognition. In this situation, the eigen decomposition approach is applied instead of SVD. This procedure entails determining the tensor's covariance matrix and calculating its eigen- vectors in decreasing order of the corresponding eigenvalues. Kernel principal component analysis (KPCA) is a nonlinear form of PCA that is built with a kernel function. The data set can be mapped onto a higher dimensional feature space F via a nonlinear mapping.

For certain feature spaces *F* there is a function for computing scalar products in feature spaces .Given a set of centered data $Y_k \in R^N$ [14][15]:

$$\sum_{k=1}^{M} \Phi(Y_k) = 0 \quad (6)$$

PCA diagonalized the covariance matrix in *F:*

$$C = \frac{1}{M}\sum_{i=1}^{M} \Phi(Y_i)\,\Phi(Y_i)^T \quad (7)$$

To find the eigenvalues $\lambda \ge 0$ and eigenvectors satisfying $\lambda \mathbf{v} = C\mathbf{v}$ All solutions v with $\lambda \ne 0$ must lie in the span of Φ (Y1), Φ (Y2),..., Φ (YM). Then there exist coefficients $\alpha\, i$ (i= 1, 2, ..., M) such that :

$$v = \sum_{i=1}^{M} \alpha_i \Phi(Y_i) \quad (8)$$

By defining the kernel function as

$$k(Y_i, Yj) = \Phi(Y_i).\Phi(Y_j) \quad (9)$$

and $K_{ij} = (\Phi(Y_i).\Phi(Y_j))$ i=1....M and j=1....M Then :

$$M\,\lambda K\alpha = K^2\alpha \quad (10)$$

where α is the column vector with entries α 1,..., α_M. The eigenvalue problem is solved$M\lambda\boldsymbol{\alpha} = K\boldsymbol{\alpha}$.

For the principal component extraction, the projections onto the eigenvectors v^kin *F* are needed.

$$(v^k.\Phi(Y)) = \sum_{i=1}^{M} \alpha_i^k (\Phi(Y_i).\Phi(Y)) \quad (11)$$

Where, Y as a test point and Φ(Y) is its image in *F*. With this method (as in PCA) we are:

The main benefit of PCA is that it can be used in an eigenface approach, which reduces the amount of the database needed to recognise test images. The feature vectors of the images are saved in the database, which are discovered by projecting each trained image to the set of Eigen faces obtained. To minimise the dimensionality of a huge data collection, PCA is used in conjunction with the Eigen face technique.

B. Face Recognition Techniques

1) Euclidian Distance Method: : Method of Euclidian Distance: The purpose of distance measures in classification issues is to assess the similarity or dissimilarity of any pair of items. The distance between two instances xi and xj can be expressed as: d (xi ,xj). The distance between each data object can be calculated using the Euclidean distance metric, as shown below:

$$Dist = \sqrt{\sum_{k=1}^{N} (X_{ik} - X_{jk})^2}$$

The minimum distance mean the more similarity

2) K Nearest Neighbour Method : Method of K Nearest Neighbors: The algorithm determines which points from the training set are similar enough to be considered when predicting the class for a new observation by selecting the k closest data points to the new observation and selecting the most common class among them. The following is a summary of the algorithm:

- A positive integer k, as well as a new sample, are supplied.
- We select the k records in our database which are closest to the new sample
- Then identify the most prevalent classification for these entries, and we apply it to the new sample.

C. Eigen Face Approach

Because of its simplicity, quickness, and learning ability, it is an adequate and efficient method for face recognition. Eigen faces are a set of Eigen vectors used in the Human Face Recognition problem in Computer Vision. They refer to a face recognition strategy based on appearance, which aims to capture variance in a collection of face photos and utilise this information to encode and compare images of individual faces in a holistic manner.

The Eigen faces are the Principal Components of a face distribution, or, to put it another way, the Eigen vectors of the covariance matrix of the collection of face images, where an image of N by N pixels is regarded a point in N 2 dimensional space. Previous face recognition research overlooked the issue of face stimuli, thinking that predetermined measurements were sufficient and meaningful. This shows that coding and decoding facial photos can provide information about the importance of features.These characteristics may or may not be linked to facial characteristics such as the eyes, nose, lips, and hairs. We want to extract the necessary information from a face image, efficiently encode it, then compare one face encoding to a database of similarly encoded faces. To extract the information content in a face image, one simple method is to capture the variance in a series of face images.

III. Methods Implemented

A. Dimensionality Reduction

B. Principal Component Analysis

C. 2d-Principal Component Analysis

D. Kernel Principal Component Analysis

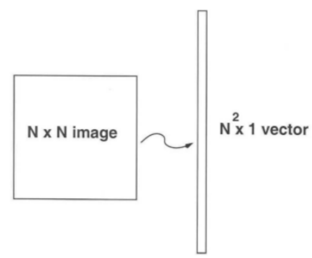
E. Eigen Face Approach

F. Eigen Values and Eigen Vectors

G. Face Recognition using above methods

IV. PCA Training Algorithm

- Let's Consider a set of m images of dimension N by N (training images).
- We first convert these images into vectors of size N2 such that:

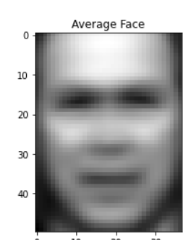


$$x_1, x_2, x_3 ... x_m$$

- Now we calculate the average of all these face vectors and subtract it from each vector

$$\psi = \frac{1}{m} \sum_{i=1}^{m} x_i$$

$$a_i = x_i - \psi$$



- Now we take all face vectors so that we get a matrix of size of N2 * M.

$$A = |a_1 \quad a_2 \quad a_3 \quad .... \quad a_m|$$

- By multiplying A with AT, we can now find the Covariance matrix. AT has the dimensions M * N2 because A has the dimensions N2 * M. When we multiply this, we get a matrix of N2 * N2, which gives us N2 eigenvectors of N2 size, which is not computed efficiently. As a result, we multiply AT and A to get our covariance matrix. This results in a M * M matrix with M (assuming M ——N2) eigenvectors of size M.

  Cov = A' . A
- In this step we calculate eigen values and eigenvectors of above covariance matrix using the formula below.

$$A^T A \nu_i = \lambda_i \nu_i$$

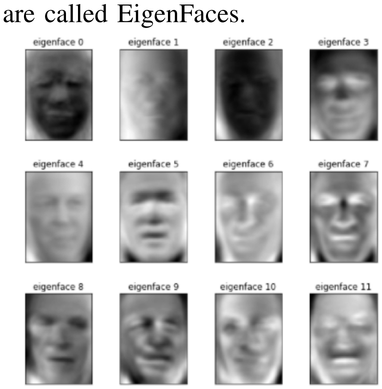
$$A A^T A \nu_i = \lambda_i A \nu_i$$

$$C' u_i = \lambda_i u_i$$

  where, C'=A.A' and ui=Avi
- It can be deduced from the above assertion that C' and C have the same eigenvalues, and their eigenvectors are connected by the equation ui = A.vi. As a result, the M biggest eigenvalues (and eigenvectors) of C' are determined by the M eigenvalues (and eigenvectors) of the covariance matrix.
- Using the formula ui = A.vi, we can now calculate the Eigenvector and Eigenvalues of this reduced covariance matrix and map them into the C'. We now choose the K C' eigenvectors that correspond to the K biggest eigenvalues (where K is less than M). The size of these eigenvectors is N square.
- We used the eigenvectors obtained in the previous phase in this step. We take the normalised training faces xi (face – average face) and express each face vector in the linear of the best K eigenvectors (as shown in the diagram below).

$$\mathrm{x}_i - \psi = \sum_{j=1}^{K} w_j u_j$$

  These uj are called EigenFaces.



- In this step, we take the coefficient of eigenfaces and represent the training faces in the form of a vector of those coefficients.

$$x_i = \begin{bmatrix} w_1^i \\ w_2^i \\ w_3^i \\ . \\ . \\ w_k^i \end{bmatrix}$$

V. Testing/Detection Algorithm

We must first preprocess an unknown face y so that it is centred in the image and has the same dimensions as the training face. Now we subtract the average face psi from the face.

$$\phi = y - \psi$$

We project the normalised vector into eigenspace to acquire the linear combination of eigenfaces.

$$\phi = \sum_{i=1}^{k} w_i u_i$$

From the aforementioned projection, we construct the coefficient vector as follows:

$$\Omega = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ . \\ . \\ w_k \end{bmatrix}$$

Subtract the vector produced in the previous step from the training image to get the shortest distance between the training and testing vectors.

$$e_r = min_l \left\| \Omega - \Omega_l \right\|$$

It is recognised with the l face from the training image if this er is smaller than the tolerance level Tr; else, it is not matched with any of the faces in the training set.

VI. 2-Dimensional PCA Algorithm

The techniques of image representation and feature extraction are widely employed in the face recognition process. When an algorithm's input data is too vast to handle and is suspected of being infamously redundant, the data is translated into a smaller representation set of features. Feature extraction is the process of transforming raw data into a set of features. It is assumed that the features set will extract the required information from the input data to accomplish the intended task using this smaller representation instead of the full-size input if the features extracted are correctly chosen. In two dimensional PCA, the image matrix A is transformed onto the X matrix using linear transformation given by:

Y = AX

So, Y is a projected vector of image A. Y is also called a projected feature vector. The total scatters of the projected samples can be characterized by the trace of the covariance matrix of the projected feature vectors. So the idea is to maximize the following:

J (X) = tr(Sx)

Where Sx denotes the covariance matrix of the projected feature vectors of the training samples and tr(Sx) denotes the trace of Sx. The physical significance of maximizing the criterion in Eq.(2) is to find a projection direction X, onto which all samples are projected so that the total scatter of the resulting projected samples is maximized. The covariance matrix Sx is given by

$$S_x = E(Y-EY)*E(Y-EY)^T = E[(A-EA)X]*E[(A-EA)X]^T \quad (3)$$

Hence

$$J(x) = X^T E[(A-EA)*(A-EA)^T]X \quad (4)$$

For given a set of training image A(1), A(2),..., A(M).

$$J(X) = X^T\left[\sum_{n=1}^{M}(A(i)-\bar{A})(A(i)-\bar{A})^T\right]X \quad (5)$$

Where is the average of the training images. Now, let

$$G_t = E\left[(A(i)-\bar{A})^T(A(i)-\bar{A})\right] \quad (6)$$

The matrix Gt is called the image covariance (scatter) matrix. It is easy to verify that Gt is an n*n nonnegative definite matrix from its definition, here Gt is computed directly from the training image. Suppose that there are M training image samples in total, the jth training image is denoted by an m*n matrix Aj(j=1,2,3,. . . , M), and the average image of all training samples is denoted by. Then, Gt can be evaluated by

$$G_t = \frac{1}{M}\sum_{j=1}^{M}(A(j)-\bar{A})^T(A(j)-\bar{A}) \quad (7)$$

Alternatively (2) can be written as

$$J(X) = X^T G_t X \quad (8)$$

Where X is a column vector with a single unit. The generalized total scatters criteria is the name of this criterion. The optimal projection axis is the unitary vector X that maximizes the criterion. Intuitively, this indicates that after projecting an image matrix onto X, the total scatters of the projected samples are maximized. The best axis for projection Xopt is the unitary vector that maximizes J(X), i.e., the Gt eigenvector with the largest eigenvalue. In general, having only one ideal projection axis is insufficient. We normally need to choose a set of projection axes, X1;...; Xd, while keeping orthonormal restrictions in mind and maximizing the J(x) criterion.

$$\{X_1, X_2, \dots, X_d\} = \arg\max(J(X)) \quad (9)$$

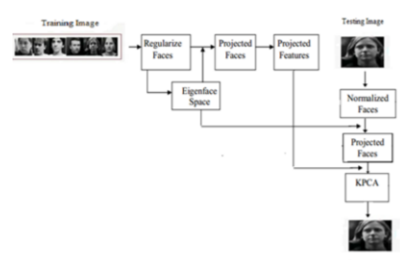
And

$$X_i^T X_i = 0 \quad ; i \neq j,\ i, j = 1, 2, \dots, d. \quad (10)$$

The optimal projection axis, X, X,..., X 1 2 d are the orthonormal eigenvectors of Gt corresponding to the first d largest eigenvalues. The optimal projection vectors of 2DPCA, X, X,..., X 1 2 d are used for feature extraction. For a given image sample A, let

Yk = A Xk

Then, we obtain a family of projected feature vectors, Y1; . . .; Yd, which is called the principal component of the sample image A. The principal component vectors obtained are used to form an m*d matrix B= [Y1; . . .; Yd], which is called the feature matrix or feature image of the image sample A. The feature vector generated is used to form a Feature matrix and the same will be used for Classification.



VII. Programming Language

We are using Python to create a face recognition project using pca, 2dpca, and kpca, which includes training, testing,

Fig. 3. Face Images in different angles and various variations in face

and finally the approach to accuracy, precision, and image match.

VIII. Source Code

In terms of the source code, we'll have a dataset in image form that we'll use to create or prepare another dataset in which we'll take a 64 by 64 pixel image and separate each pixel so that it's arranged in a row, giving us 4096 pixels and 4096 columns.

We do the same thing for all the image and then we train a model and pca will be applied for that prepared dataset and we test a model and pca will be applied for that prepared dataset . We repeat pca 2 times so that the accuracy and various parameters are specified upto what percentage of success we have reached by building that respective model.

So , this is the overall approach to the source code for this respective project.

A. Algorithm for PCA

1) *Step 1: Read dataset and visualize it:*
2) *Step 2: Split Dataset into training and testing:*
3) *Step 3: Perform PCA.:*
4) *Step 4: Project Training data to PCA:*
5) *Step 5: Initialize Classifer and fit training data:*
6) *Step 6: Perform testing and get classification report:*

B. Algorithm for 2d-PCA

1) *Step 1: Read dataset and visualize it:*
2) *Step 2: Split Dataset into training and testing:*
3) *Step 3: Perform 2d-PCA.:*
4) *Step 4: Project Training data to 2d-PCA:*
5) *Step 5: Initialize Classifer and fit training data:*
6) *Step 6: Perform testing and get classification report:*

C. Algorithm for k- PCA

1) *Step 1: Read dataset and visualize it:*
2) *Step 2: Split Dataset into training and testing:*
3) *Step 3: Perform k-PCA.:*
4) *Step 4: Project Training data to k-PCA:*
5) *Step 5: Initialize Classifer and fit training data:*
6) *Step 6: Perform testing and get classification report:*

IX. Dataset

X. Experiments Planned

A. Face Recognition using PCA

B. Face Recognition using 2d-PCA

C. Face Recognition using kPCA

Therefore , we are planning to implement all the above experiments in our project for biometric face recognition using mathematical dimensionality reduction processes where each process is better than the other in python.

XI. Links to the Code

Face Recognition Using PCA
Face Recognition Using 2D-PCA
Face Recognition Using K-PCA

References

[1] Rafael Gonzalez and Richard Woods. Digital Image Processing. Addison Wesley, 1992.

[2] M. A. Turk and A. P. Pentland. Eigenfaces for Recognition. Journal of Cognitive Neuroscience, 3(1):71 – 86, 1991.

[3] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 91, pages 586 – 591, 1991.

[4] G.M.Beumer, Q. Tao, A.M. Bazen and R.N.J. Veldhuis, A landmark paper in face recognition, 2006

[5] Rafel C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson Education, Six Indian Reprint, 2001

[6] Hui Kong, Xuchun Li, Lei Wang, Eam Khwang Teoh, Jian-Gang Wang, Ronda Venkateswarlu "Generalized 2D Principal Component Analysis", IEEE International Joint Conference on Neural Network (IJCNN), Montreal, Canada, 2005.

[7] F. Jalled, "Face recognition machine vision system using eigenfaces," 2017.