International Journal of Formal Sciences: Current and Future Research Trends (IJFSCFRT)
ISSN: 2790-7945

https://ijfscfrtjournal.isrra.org/index.php/Formal_Sciences_Journal/index

A Model for Face Recognition using EigenFace Algorithm

Vincent Mbandu Ochango*

Murang'a University of Technology, 75, Murang'a and 10200, Kenya

Email: ochangovincent@gmail.com

Abstract

The use of a computer to recognize a person by the means of their face is what is known as face recognition in artificial intelligence. The term biometrics is an umbrella term that includes face recognition as well as signature, fingerprint, eye scanning, gait, and palm print recognition. The principal component analysis technique was used in this paper to extract distinctive features from the faces which are matched with other faces stored in the database and predictive results indicated which faces were recognized and the ones that were not recognized. The accuracy of these techniques was calculated and the principal component analysis technique was found to be 86.3636% accurate and it was concluded that the technique performs better when it comes to face recognition.

Keywords: Feature Descriptor; Covariance Matrix; Eigenface; eigenvector; Linear Algebra.

1. Introduction

Nowadays problems are being solved in the field of computer vision such as medical imaging, surveillance, and face recognition through various techniques. This research aims to highlight the application of linear algebra in the field of computer vision and face recognition. Computer Vision is the same way people use their eyes and brains to sense and see the world. It uses algorithms to analyze and understand images and be able to classify images based on the information it has collected from images [1]. Linear Algebra is a concept that has been used in this paper since the techniques used for face recognition mostly use the concept of linear Algebra. The concept is widely used in the field of computer vision specifically in the area of face recognition. This report presents the idea of face recognition using Eigen and fisher's faces and gives the basic idea of applications of linear algebra in the field of computer vision. The scope of this work is, to work on a technique to detect or recognize faces by using the method of Eigen and Fisher's faces. This is a very well-formulated application of linear algebra in the field of computer vision [2].

----

* Corresponding author.

The researcher applied face recognition technique to the face dataset, the dataset was divided into training and testing. The training dataset consisted of 60 faces while the testing dataset consisted of 44 faces. Both the faces for the training and testing dataset consisted of images of pgm file extension. The researcher first tried dimensionality reduction techniques on the dataset using principal component analysis (PCA) and linear discriminant analysis (LDA). The research visualized and explored the spectrum of uncovered Eigen and Fisher's faces. The research then explored the effect of varying parameters and compositions of different techniques on overall facial recognition performance, including varying the number of eigenvectors such as limiting eigenvectors, removing top eigenvectors found in PCA, comparing PCA, LDA, and dimensionality reduction in terms of accuracy and classification performance [3].

2. Related Work

Security concern is one of the major issues that need a lot of research hence this has motivated a lot of research in face recognition to be able to identify people easily. Different techniques are being used for extracting features from faces and be able to use these features to identify and classify faces uniquely. An experiment of face recognition techniques is what is being studied in this research to be able to recommend the technique to be used based on the experimental results. Some of the face recognition techniques widely used in the field of computer vision include Eigen Faces, Fisher's Faces, Neural Network, Dynamic Link Architecture, Hidden Marvok Model, Geometrical Feature Matching, Template Matching, etc.

The most widely used face recognition technique is Eigen Faces which is also known as the Principal Component Analysis method. According to Sirovich 2019, Eigen Faces is a method that is used to extract only relevant features from faces and these relevant features are the ones that are used to reconstruct faces [4]. Petland in 2017 obtained 85% right categorization of faces by using Eigenfaces. The method usually extracts features from the face and the resultant features are in the form of the feature vectors hence it is considered as the principal component. The covariance matrix is the one that is usually used to generate the feature vector that is used to represent the face. The feature vectors are known as the Eigen Vectors are the ones that are usually used to differentiate one image from another. In 2014 Petland et al attained recognition of 0.95 on the PERET image dataset which had 7562 images. The experiment used a less sensitive approach of extracting Eigen features from eyes, cheeks, mouth, and nose and using the features for face recognition unlike the other method of using Eigen Faces [5].

Neural Network is one of the methods that is used in face recognition, Sung et al in 2015 used Convolution Neural Network and Multilayer Perceptron for face recognition. In their experiment, they used 400 images for the ORL database and they achieved a face recognition rate of 0.962 with Convolution Neural Network. According to their experiment, it took 0.5 seconds to extract features from the dataset and training the model from the features extracted from the training dataset took four hours [6].

The hierarchy of computer vision would be a four-staged process as shown below

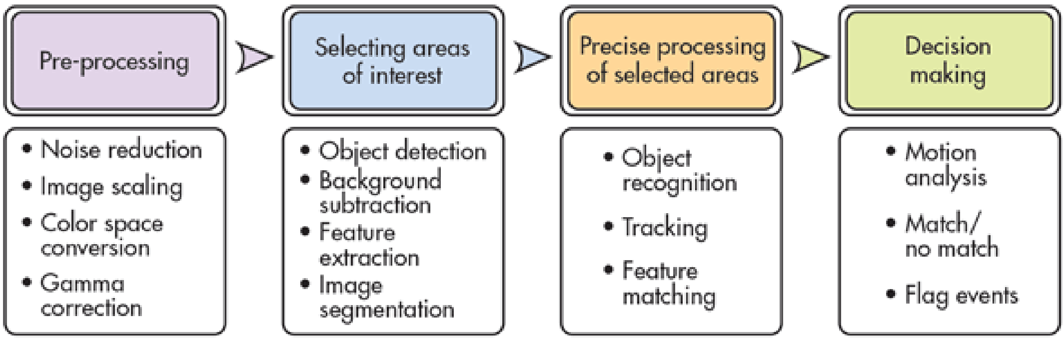

Figure 1: Hierarchy of Computer Vision

The approaches have been made in many different ways to perform the task of face recognition like the one where automated face recognition is used. These methods of performing the task of face recognition, each one has their advantages and disadvantages. It is our part to choose the best available method.

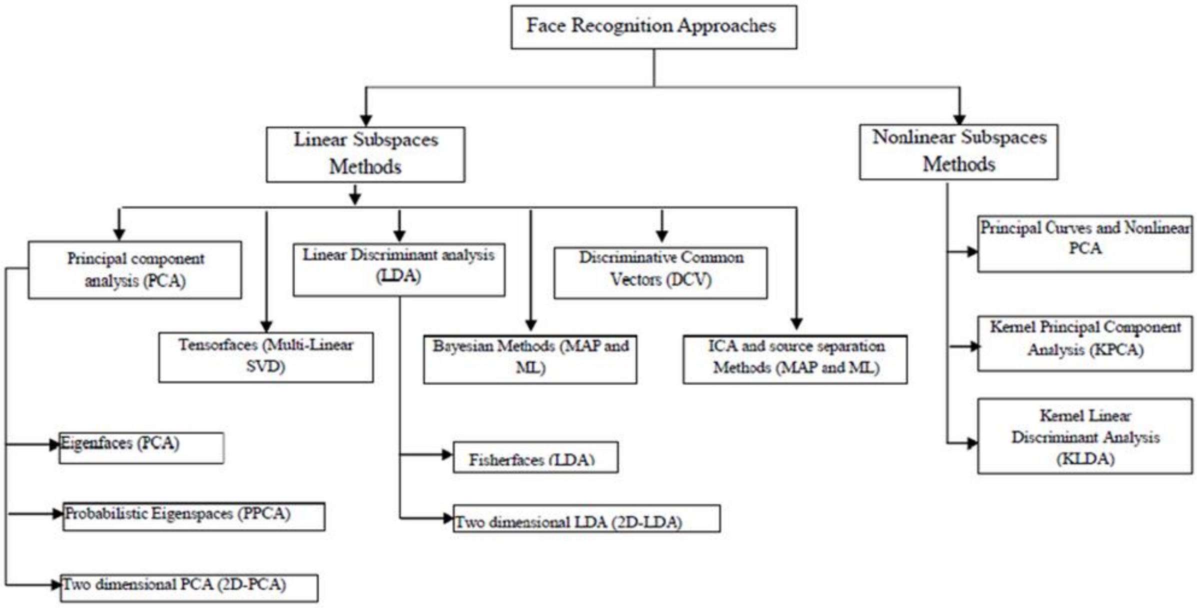

Figure 2: Face Recognition Approaches

The face Recognition technique can be done by considering the variation of one face of a person from the face of the other person in terms of ratio and that is what exactly Fisher's face method does. Methods used to recognize faces have weaknesses and advantages hence face recognition is usually affected by certain factors that in turn affect the performance of these techniques. Furthermore, accuracy usually varies depending on some face recognition techniques, and the best way one can try to reduce these factors that affects face recognition is to use the hybrid approach [7].

3. Methodology

3.1 Data Collection and Preprocessing

The face dataset consisted of 104 images and training dataset consisted of 60 images and the testing dataset consisted of 44 images and the image were of type PGM since they were edited using adobe photoshop. The image dataset was used as a secondary dataset obtained from Github. To read the images we used anaconda IDE that supports python programming language to import imread library from matplotlib.image.

3.2 Train/Test Split

The researcher used an image dataset that was split into two partitions: train and test, housed in different folders to ensure that our test sets are untainted by downstream processing. It was ensured that the training and test sets are sampled from the same distribution by taking an equal proportion of samples from each folder.

3.3 Face Recognition Techniques

i. Computing Eigen Faces

- **Main idea behind eigenfaces**

- Suppose Γ is an N^2x1 vector, corresponding to an NxN face image I.

- The idea is to represent Γ (Φ=Γ - mean face) into a low-dimensional space:

$$\hat{\Phi} - mean = w_1 u_1 + w_2 u_2 + \cdots w_K u_K \ (K \ll N^2)$$

- **<u>Step 1:</u>** Obtain face images I_1, I_2,………I_M (Training images).
- **<u>Step-2:</u>** Represent every image I_i as Γ_i .
- **<u>Step-3:</u>** Compute average face vector Ψ.

$$\Psi = \frac{1}{M} \sum_{i=1}^{M} \Gamma_i$$

- **<u>Step-4:</u>** Subtract the mean face.

$$\Phi_i = \Gamma_i - \Psi$$

- **<u>Step-5:</u>** Compute the covariance matrix C.

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T = AA^T \quad (N^2 \text{x} N^2 \text{ matrix})$$

where $A = [\Phi_1\ \Phi_2 \cdots \Phi_M] \quad (N^2 \text{x} M \text{ matrix})$

- **<u>Step-6:</u>** Compute the eigen vectors u_i of AA^T

The matrix AA^T is very large --> not practical !!

Step 6.1: consider the matrix A^TA (MxM matrix)

Step 6.2: compute the eigenvectors v_i of A^TA

$$A^TAv_i = \mu_i v_i$$

What is the relationship between us_i and v_i?

$A^TAv_i = \mu_i v_i \Rightarrow AA^TAv_i = \mu_i Av_i \Rightarrow$

$CAv_i = \mu_i Av_i$ or $Cu_i = \mu_i u_i$ where $u_i = Av_i$

Thus, AA^T and A^TA have the same eigenvalues and their eigenvectors are related as follows: $u_i = Av_i$!!

Note 1: AA^T can have up to N^2 eigenvalues and eigenvectors.

Note 2: A^TA can have up to M eigenvalues and eigenvectors.

Note 3: The M eigenvalues of A^TA (along with their corresponding eigenvectors) correspond to the M *largest* eigenvalues of AA^T (along with their corresponding eigenvectors).

Step 6.3: compute the M best eigenvectors of AA^T: $u_i = Av_i$

(**important:** normalize u_i such that $\|u_i\| = 1$)

ii. **Representing faces on this basis**

- Each face ϕ_i (minus the mean) can be represented as a linear combination of the best K eigenvectors

$$\hat{\Phi}_i - mean = \sum_{j=1}^{K} w_j u_j, \quad (w_j = u_j^T\Phi_i)$$

(we call the u_j's *eigenfaces*)

- Each normalised face ϕ_i is represented in the vector:

$$\Omega_i = \begin{bmatrix} w_1^i \\ w_2^i \\ \cdots \\ w_K^i \end{bmatrix}, \quad i = 1, 2, \ldots, M$$

iii. **Recognition of faces**

- Given an unknown face image Γ (centered and of the same size like the training faces) follow these steps:

Step 1: normalize Γ: $\Phi = \Gamma - \Psi$

Step 2: project on the eigenspace

$$\hat{\Phi} = \sum_{i=1}^{K} w_i u_i \quad (w_i = u_i^T \Phi)$$

Step 3: represent Φ as: $\Omega = \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_K \end{bmatrix}$

Step 4: find $e_r = \min_l \|\Omega - \Omega^l\|$

Step 5: if $e_r < T_r$, then Γ is recognized as face l from the training set.

- The distance e_r is called distance within the face space (difs)

Comment: we can use the common Euclidean distance to compute e_r, however, it has been reported that the *Mahalanobis distance* performs better:

$$\|\Omega - \Omega^k\| = \sum_{i=1}^{K} \frac{1}{\lambda_i} (w_i - w_i^k)^2$$

(variations along all axes are treated as equally significant)

The below image provides a flowchart of the Eigen Faces algorithm

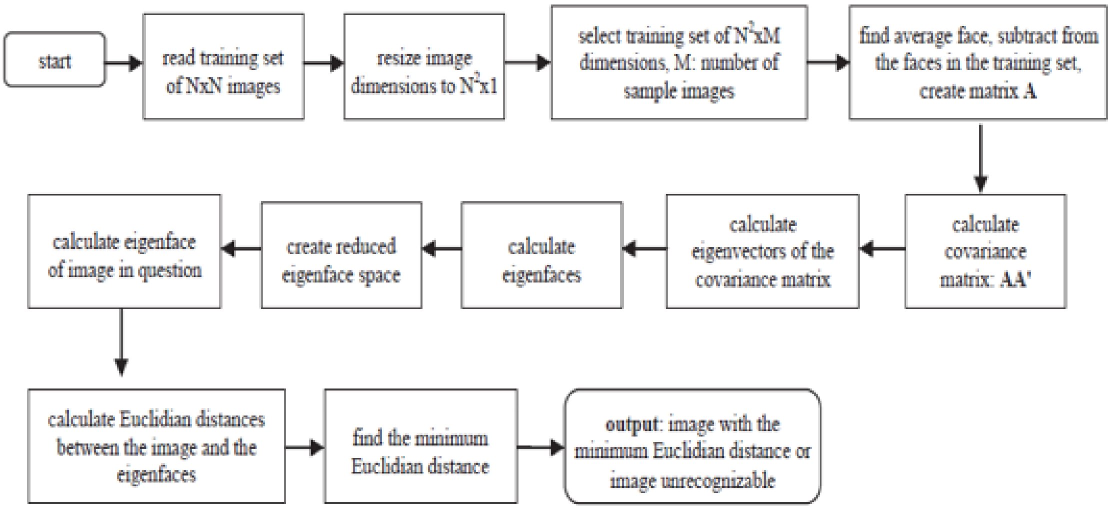

Figure 3: Flowchart of the Eigen Faces algorithm

3.4 Dimensionality Reduction

Principal Component Analysis (PCA)

The primary use of PCA in this research is for dimensionality reduction. We treat each image as a flattened 1-dimensional vector and find the covariance matrix on the matrix of images, where each image is a column vector. We then find the top k eigenvectors of this covariance matrix, in order of the magnitude of their eigenvalues. These eigenvectors represent the eigenfaces of our training dataset. The images were then projected onto these eigenfaces, approximately treating all faces as a linear combination of these eigenvectors. This gave a dimensionality reduction when we limited the number of eigenvectors we choose to use to represent each image.

Pre-processing (Mean Centering)

To properly apply PCA, it was first ensured that image vectors are mean-centered. The summation of all image vectors in the training dataset was done and divided by the magnitude of the training dataset to get the average image. Every image in the training set is subtracted by this average. This ensures that the training set is centered at a mean 0. The images in the test dataset are all subtracted by the same mean image, to ensure the same transformations are applied on the test dataset. As a sanity check, the average images are visualized from both the training and test sets and it is found out that they are nearly identical.

4. Experimental Results and Discussion

4.1 Mean Face

The average face is obtained by converting all the images in the training dataset into an array. The array of each image is then summed up to get the total summation of all the images in the dataset. The final summation of all the arrays is then divided by the total number of images and in our case, we had 60 images for the training dataset. The mean array is then plotted using matplot library as shown in figure 4.

Figure 4: Mean Face

4.2 Mean Centering

Every image in the training set was subtracted by the mean face. This ensured that the training set is centered at a mean 0. The images in the test dataset were all subtracted by the same mean image, to ensure the same transformations are applied on the test dataset.

4.3 Covariance Matrix

The covariance matrix of the normalized training dataset is computed. The matrix helped us determine how similar the variances of the face features are.

```
Covariance matrix of X:
[[25.76481507  5.67350101  2.36306105 ... -1.54762144 -0.0806879
  -5.17057477]
 [ 5.67350101 21.22452574  5.01228367 ...  0.69290791  3.43353284
   0.23688211]
 [ 2.36306105  5.01228367 33.29665458 ... -0.79807283 -0.24953619
   0.14335801]
 ...
 [-1.54762144  0.69290791 -0.79807283 ... 15.58735801  9.63566816
   7.62289475]
 [-0.0806879   3.43353284 -0.24953619 ...  9.63566816 23.14454622
   9.90017353]
 [-5.17057477  0.23688211  0.14335801 ...  7.62289475  9.90017353
  17.51264716]]
```

Figure 5: Covariance Matrix of Normalised Training data set

4.4 Eigenfaces

The eigenvectors of the covariance matrix of all images were generated in the training set and sorted by descending magnitude. The top 8 rendered eigenvectors resemble faces. The next few rows of 9 also vaguely have facial characteristics, but by row 4 and beyond, the facial characteristics have severely degraded.

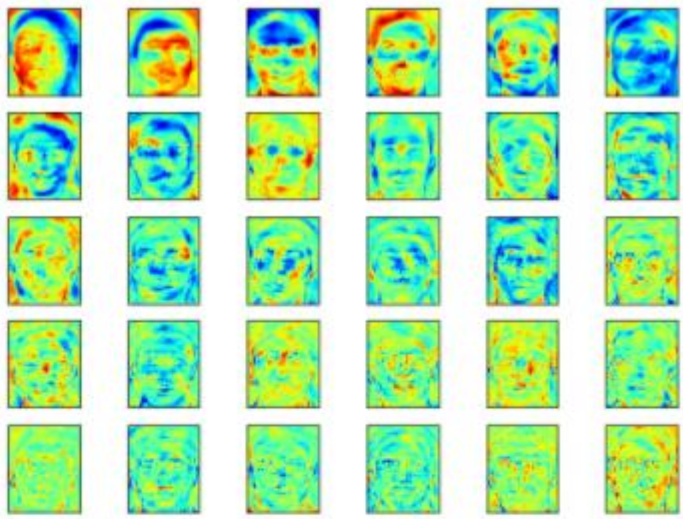

Figure 6: Eigen Faces

4.5 Testing all the Images

The testing image data set was normalized by subtracting the mean face from the testing image vector. The testing images were then compared with the one in the training dataset to see if the matching can be achieved and if yes how many unknown faces matched with the one in the training dataset. The 44 faces from the testing dataset were matched with the ones in the training dataset and out of 44 faces, 38 faces matched and the rest did not match, which resulted in an accuracy of 86.3636%. The eigenface approach helps in face recognition hence making the identification of faces easier and faster even though it does not give 100% accuracy, it can still be used to recognize people's faces since it balances the tradeoff between computational complexity and accuracy

5. Conclusion

The rejection rate can be reduced by making the eigenface technique work at a very high accuracy which most methods lack. The accuracy of the method was calculated and out of 44 faces tested only 38 matched with the training dataset which resulted in an accuracy score of 86.3636%. The face triangle information which is always important can be incorporated with the eigenface technique in the future to make the technique more accurate. The information theory and linear algebra are the concepts behind the eigenface face recognition technique, this makes the method approximate the set of known face images using fewer face features which leads to a balanced tradeoff between computational complexity and accuracy. The eigenface approach helps in face recognition hence making the identification of faces easier and faster even though it does not give 100% accuracy, it can still be used to recognize people's faces since it balances the tradeoff between computational complexity and accuracy. In future experiments, the method can be compared with other face recognition techniques to come to a conclusion on which method can be best used when face recognition problems arise. In conclusion, the highest eigenvalues from the eigenfaces can be used to reconstruct faces which in turn helps in face recognition. The more the training faces are used with the eigenface approach the more the technique becomes accurate. The euclidean distance is obtained by subtracting the test face weight from the train face weight. During prediction, the euclidean distance must be less than the threshold for a face to be classified to a certain class else the face is classified as an unknown person.

References

[1] Bahurupi, S. P. (2018). Principal component analysis for face recognition. International Journal of Engineering and Advanced Technology, 91-94.

[2] Frey, B. J., Colmenarez, A., & Huang, T. S. (2018, June). Mixtures of local linear subspaces for face recognition. In Proceedings. 20188 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231) (pp. 32-37). IEEE.

[3] Joseph, J. &. (2016). Automatic attendance management system using face recognition. International Journal of Science and Research (IJSR), 327-330.

[4] Meethongjan, K. &. (2017). A Summary of literature review: Face Recognition. In Postgraduate annual research seminar.

[5] Sahoolizadeh, H. &. (2018, September). Face recognition using eigen-faces, fisher-faces and neural networks. In 2018 7th IEEE International Conference on Cybernetic Intelligent Systems (pp. 1-6). IEEE.

[6] Sandhu, P. S. (2019). Face Recognition Using Eigen face Coefficients and Principal Component Analysis. International Journal of Electrical and Electronics Engineering, 498-502.

[7] Sharif, M. M. (2018). A survey: face recognition techniques. Research Journal of Applied Sciences, Engineering and Technology, 4979-4990.

[8] Turk, M. A. (2019, January). Face recognition using eigenfaces. In Proceedings. 2019 IEEE computer society conference on computer vision and pattern recognition (pp. 586-587). IEEE Computer Society.

[9] Wen, Y., He, L., & Shi, P. (2012). Face recognition using difference vector plus KPCA. Digital Signal Processing, 22(1), 140-146.

[10] Zhang, C., Zhou, Z., Sun, H., & Dong, F. (2012, May). Comparison of three face recognition algorithms. In 2012 International Conference on Systems and Informatics (ICSAI2012) (pp. 1896-1900). IEEE.