Facial Recognition: Eigenfaces

Project Report

MAT161: Applied Linear Algebra

12 May, 2023

Arnav Aditya [2210110189]

Index

Acknowledgement

We thank Professor Santosh Singh (Professor, Department of Mathematics, School of Natural Sciences, Shiv Nadar Institution of Eminence) for the assignment of the project and simplification of the rules and procedure for making this report as well as the program. We were truly inspired by the concepts he delivered in his lectures, particularly those pertaining eigenvalues and eigenvectors, which made it much easier to grasp the concepts related to this project.

Through this project, we understood the real-world applications of the concepts we learn in the classroom, and how they can be effectively implemented in technology. This motivated all of us to further pursue mathematics and learn the concepts with sincerity and diligence, owing to their immense real-world applications. Lastly, we are indebted to various internet resources (mentioned fully in the bibliography and tools section of the report) as we understood and learnt from them in the making of the project.

Introduction

The face is crucial in social interactions and can convey identity and emotions. Humans have remarkable abilities to recognize faces, even after years of separation, and this ability is robust despite changes in visual stimuli. Computational models of face recognition can contribute to both theoretical insights and practical applications, such as criminal identification, security systems, and human-computer interaction. However, developing a computational model of face recognition is challenging due to the complexity, multidimensionality, and meaningfulness of facial stimuli. Based on the research done by *Mathew Turk* and *Alex Pentland*, including their research paper titled *"Eigenfaces for Recognition"*, we tried to develop a fast, simple, and accurate computational model of face recognition that is biologically implementable and compatible with the physiology and psychology of face recognition. Our approach is based on an information theory that decomposes face images into a small set of characteristic feature images called "*eigenfaces*". Recognition is achieved by projecting a new image into the subspace spanned by the eigenfaces and comparing its position with the positions of known individuals. This framework allows for automatic learning and recognition of new faces and is insensitive to small or gradual changes in the face image. This approach has advantages over other face recognition schemes in its speed, simplicity, and learning capacity.

Methodology

The research paper "*Eigenfaces for Recognition*" proposed a novel method for face recognition using a statistical approach. The authors introduced the concept of eigenfaces, which are eigenvectors derived from the covariance matrix of face image data. The primary steps involved in the eigenface approach are:

I. *Collection of Face Image Data*: In the first step of the eigenfaces method, a dataset of face images is collected. This dataset typically consists of grayscale images of individuals' faces, with each image representing a different individual.
To create the dataset, a large number of images of different individuals are captured under controlled conditions. These images can be obtained from various sources, such as databases or by capturing images from a camera. The dataset should be large enough to cover a diverse range of facial features, expressions, lighting conditions, and angles.

II. *Pre-processing*: Pre-processing is an essential step in the eigenfaces method for facial recognition. It involves several techniques to improve the quality and consistency of the dataset, making it easier for the algorithm to recognize faces accurately.
The first step in pre-processing is to remove noise from the images. This can be done by applying filters to the images to reduce blur, smoothen the edges, and enhance the sharpness of the facial features. Noise can arise from several sources, such as the camera's sensor or

compression during image storage and transmission. Removing noise can help improve the accuracy of the algorithm.

III. *Normalisation:* The next step in pre-processing is to normalize lighting conditions across all images in the dataset. Lighting conditions can vary significantly in different environments, leading to shadows, reflections, and changes in brightness and contrast. Normalizing lighting conditions involves adjusting the brightness and contrast of the images to a standard level, making them consistent across the dataset. This step helps reduce the impact of lighting variations on facial recognition accuracy.

IV. *Alignment:* Alignment of facial features is another crucial pre-processing step. This involves ensuring that the facial features in all images are aligned consistently. Misalignment can occur due to changes in the pose, angle, or position of the face in the image. Alignment can be done using various techniques, such as detecting facial landmarks, such as the eyes, nose, and mouth, and adjusting the position and angle of the image accordingly.
Finally, the images can be converted to grayscale to simplify the processing and reduce the size of the dataset. Greyscale images contain only one colour channel, making them easier to handle and process than coloured images.

V. *Dimensionality Reduction*: Dimensionality reduction is a critical step in the eigenfaces method for facial recognition. It involves reducing the dimensionality of the face image data while retaining the most significant features of the faces. The reduction in dimensionality helps in reducing the computational complexity and storage requirements of the algorithm.

VI. *Principal Component Analysis* (PCA) is a widely used technique for dimensionality reduction in eigenfaces. PCA involves computing the covariance matrix of the face image data and finding its eigenvectors, which are known as eigenfaces. The eigenfaces represent the most significant features of the face images and capture the variability in the dataset.
To compute the covariance matrix, the mean face image is first calculated by averaging all the images in the dataset. The mean image is then subtracted from each image to obtain the deviation of each image from the mean. The covariance matrix is then computed from the deviations of all the images. The covariance matrix represents the statistical relationship between the different image pixels.

VII. *Eigenfaces:* The eigenvectors of the covariance matrix, or the eigenfaces, are obtained by solving the eigenvalue problem of the covariance matrix. The eigenfaces represent the most significant features of the face images and are orthogonal to each other. They are sorted by their corresponding eigenvalues in descending order, indicating their significance.
The eigenfaces with the highest eigenvalues capture the most significant features of the face images, while those with lower eigenvalues capture less important features. By retaining only the top

eigenfaces, the dimensionality of the face image data is reduced significantly, without losing critical information.

In summary, dimensionality reduction is an essential step in the eigenfaces method for facial recognition. It involves reducing the dimensionality of the face image data using PCA and computing the eigenvectors, or eigenfaces, of the covariance matrix. The eigenfaces represent the most significant features of the face images, and their reduction in number helps in reducing the computational complexity and storage requirements of the algorithm.

VIII. *Face Representation*: In eigenfaces method for facial recognition, after the dimensionality of the face image data has been reduced by computing the eigenvectors or eigenfaces, the next step is to represent each face image as a linear combination of these eigenfaces. This step involves projecting each face image onto the eigenspace spanned by the eigenfaces.

The projection of each face image onto the eigenspace is achieved by computing the dot product of the image vector with each eigenface vector. The resulting scalar values represent the weights of each eigenface in the representation of the face image. These weights capture the essential features of the face that are represented by the eigenfaces.

The face representation is the set of weights that represent each face image as a linear combination of the eigenfaces. These weights capture the essential features of the face and provide a compact representation of the image. The number of weights required to represent each image depends on the number of eigenfaces used to reduce the dimensionality of the image data.

The face representation can be used for facial recognition by comparing the weights of the face images in the eigenspace. The similarity between two face images can be determined by computing the Euclidean distance or cosine similarity between their weight vectors. The face image with the closest weight vector to a test image is considered the match.

IX. *Recognition*: The recognition process involves projecting an input face onto the eigenspace and comparing it with the eigenfaces using a distance metric, such as Euclidean distance or Mahalanobis distance.

X. *Classification*: The input face is classified by finding the closest match in the training set based on the distance metric.

Main Issues

There were a couple main issues that we had to resolve while working on this project, and we've listed them out below.

I. One problem is that the background can significantly affect recognition performance, so the input face image is multiplied by a two-dimensional gaussian window centred on the face to diminish the background.
II. Another issue is that recognition performance decreases quickly as head size or scale is misjudged, so the head size in the input image must be close to that of the eigenfaces. To counter this, we ended up using multiscale eigenfaces and in some cases scaling the input image to multiple sizes to address this problem.
III. Head orientation can also cause performance degradation, so the image can be rotated to align the head with the eigenfaces.
IV. Additionally, we characterized the distribution in face space rather than assuming it is Gaussian, and extending the system to deal with multiple views by defining limited numbers of face classes for each known person corresponding to characteristic views.

Significance and Impact

The "Eigenfaces for Recognition" paper made significant contributions to the field of computer vision and face recognition. Some key aspects of its significance and impact are:

I. *Introduction of Eigenfaces* : The paper introduced the concept of eigenfaces, which revolutionized the field of face recognition. Eigenfaces provided a computationally efficient and effective approach for representing and recognizing faces.

II. *Dimensionality Reduction* : The paper demonstrated the effectiveness of Principal Component Analysis (PCA) for reducing the dimensionality of face image data. This approach allowed for efficient storage and computation, making it feasible to apply facial recognition algorithms in real-world scenarios.

III. *Robustness and Generalization* : The eigenface method showed robustness to variations in pose, lighting conditions, and facial expressions. This ability to generalize across different facial variations contributed to the practical applicability of the approach.

IV. *Influence on Further Research* : The eigenface method presented in the paper served as the foundation for subsequent research in face recognition and computer vision. It inspired the development of more advanced techniques, such as Fisherfaces and deep learning-based approaches, which built upon the concept of eigenfaces.

Limitations

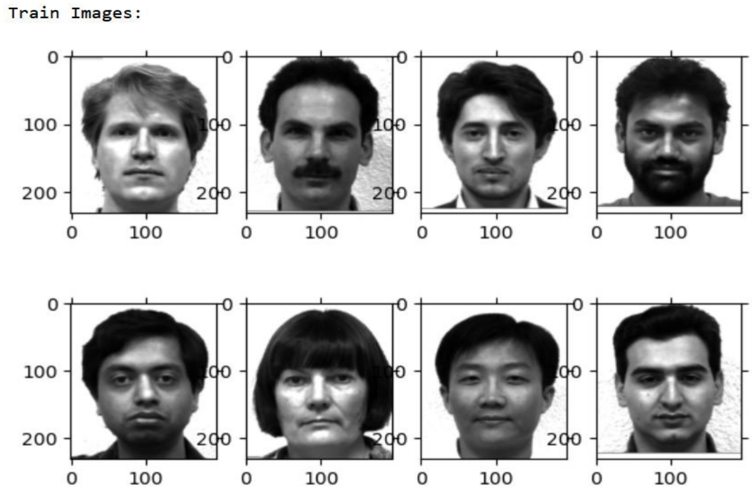
However, it is important to acknowledge certain limitations of the eigenface method. One limitation is that it treats each pixel in an image equally, disregarding the spatial arrangement of facial features. This can result in less accurate recognition in cases where local features are crucial for distinguishing between individuals.

Moreover, the eigenface method assumes linearity in the variation of face images. While this assumption holds to some extent, there are cases where non-linear variations, such as extreme facial expressions or substantial pose changes, can challenge the performance of the eigenface approach.

Additionally, eigenfaces can be computationally intensive, especially when dealing with a large number of images or a large number of eigenfaces. This can be a limitation in real-time applications, such as video surveillance or real-time identification.

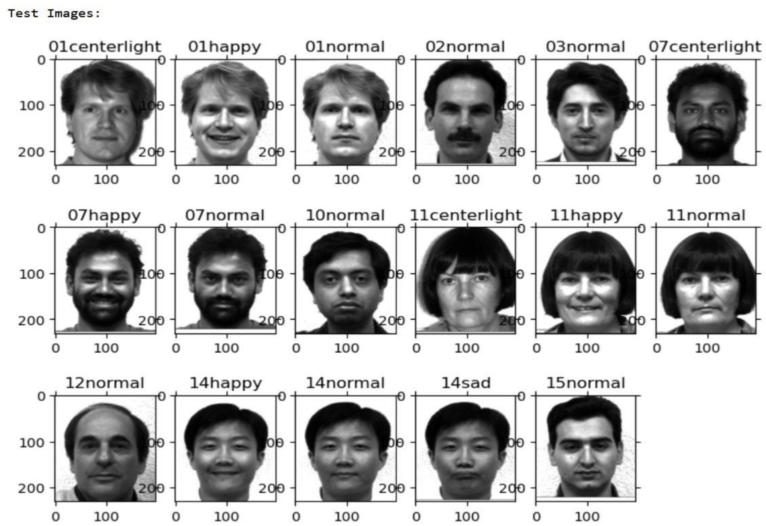
Lastly, the accuracy of eigenfaces can be greatly affected by lighting conditions during image acquisition. This means that if the lighting conditions in the new image are different from those in the training set, the recognition rate can decrease significantly.

Screenshots



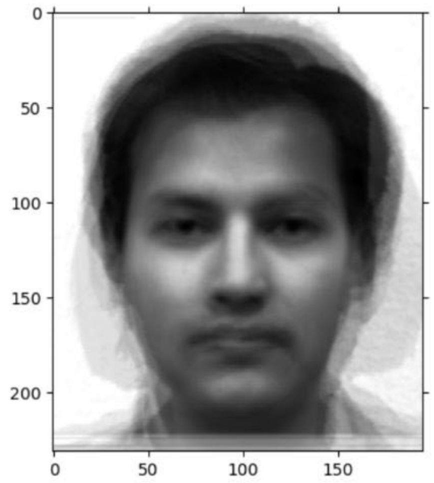
The dataset consists of a set of 17 grayscale face images (in JPG format). Each image is of dimension 195 x 231 (width x height) pixels and each pixel uses 8 bits for greyscale.

The above 7 images are the *Training Images* used to train the algorithm and make an eigenface.

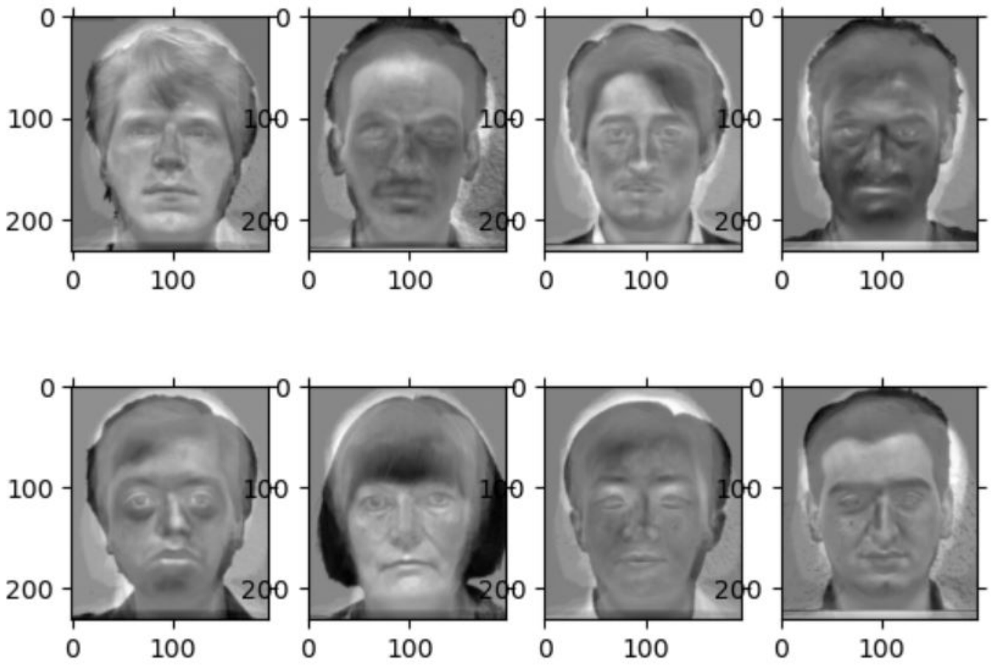


Test Images used to test the algorithm.

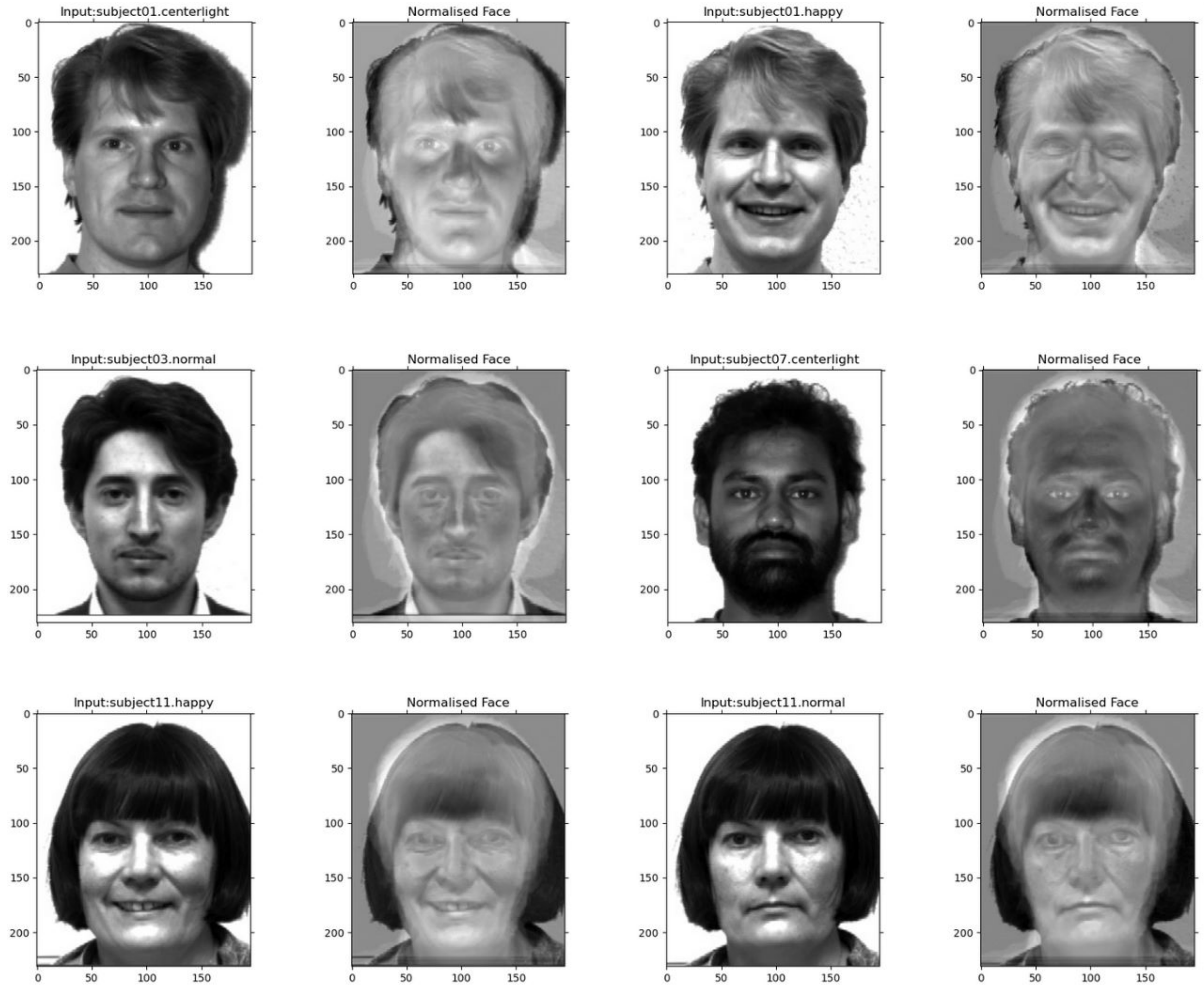
```
In [4]: mean_face = np.zeros((1,height*width))

        for i in training_tensor:
            mean_face = np.add(mean_face,i)

        mean_face = np.divide(mean_face,float(len(train_image_names))).flatten()

        plt.imshow(mean_face.reshape(height, width), cmap='gray')
        plt.tick_params(labelleft='off', labelbottom='off', bottom='off',top='off',right='off',left='off', which='both')
        plt.show()
```



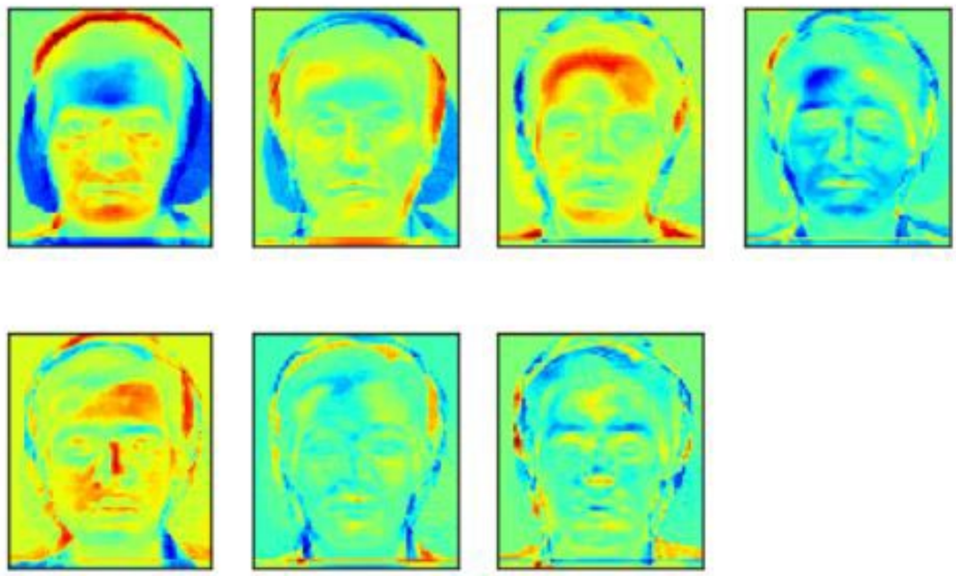
The *Mean Face* as per the algorithm.



Normalised Faces.

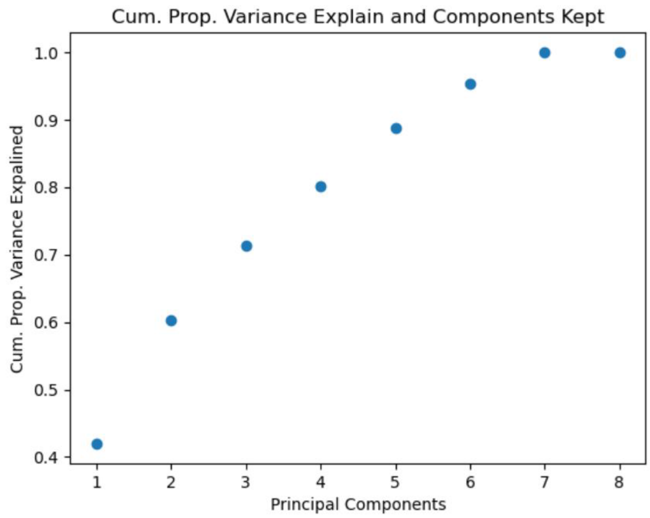


Comparison of the *Original Faces* and *Normalised Faces*.



Plotting the *Eigenfaces*.

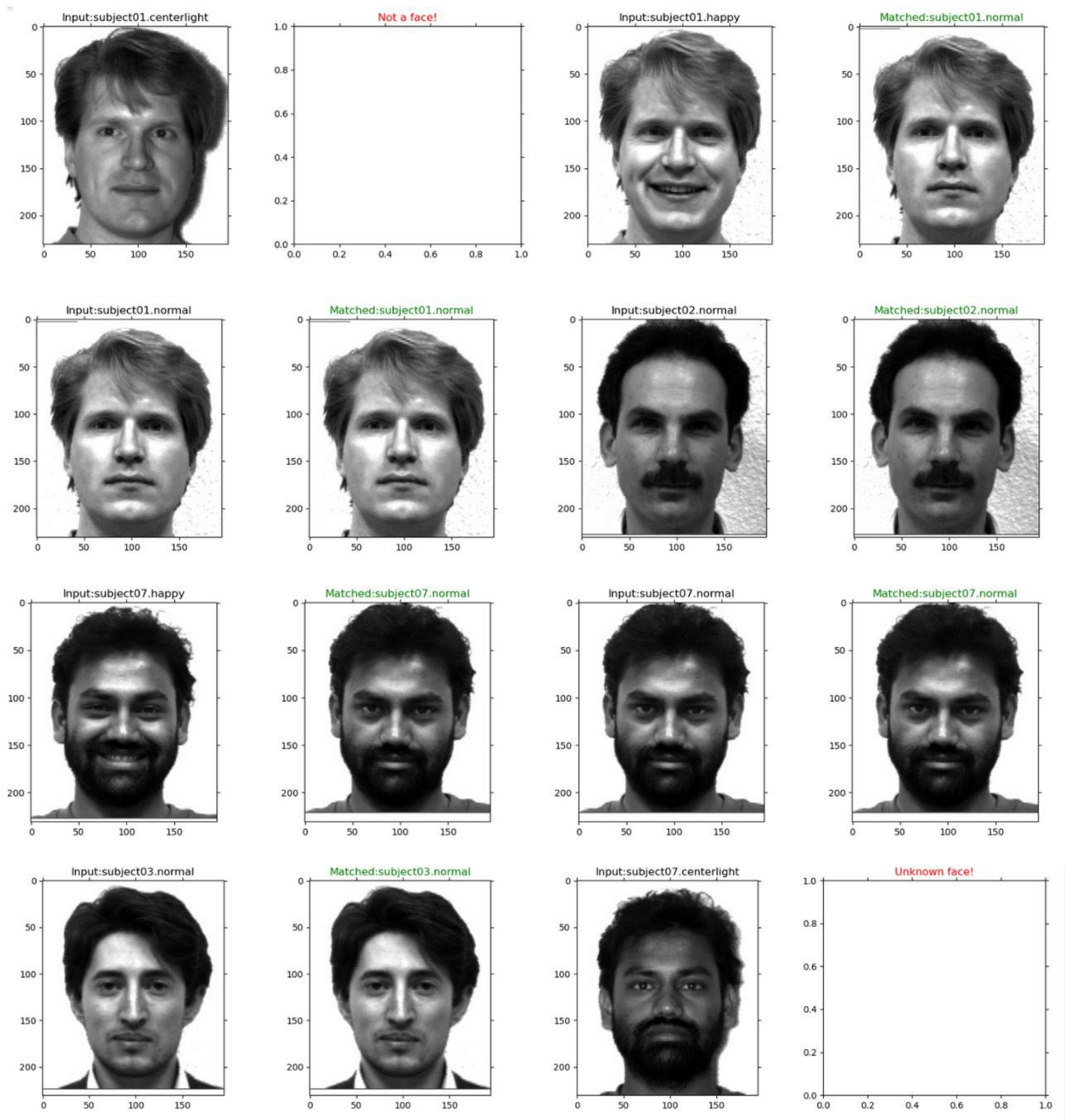
```
Cumulative proportion of variance explained vector:
[0.41996902 0.60289218 0.7140576  0.80143503 0.88729207 0.95371129
 1.         1.        ]
```



Plot of *Cumulative variance* of each *Principle Component*

```
In [7]: cov_matrix = np.cov(normalised_training_tensor)
        cov_matrix = np.divide(cov_matrix,8.0)
        print('Covariance matrix of X: \n%s' %cov_matrix)

Covariance matrix of X:
[[ 240.21425354  -54.37445049  -49.91300972 -167.04449305    6.71011608
    95.13549119   51.86167951 -122.58958706]
 [ -54.37445049  271.21637481  -39.69264581   66.46457924  -62.78262301
  -159.33970561  -96.7188796    75.22735047]
 [ -49.91300972  -39.69264581  223.857185     46.24225037  -32.6657127
  -134.1771553   -35.12864547   21.47773363]
 [-167.04449305   66.46457924   46.24225037  345.77440281  -80.00529939
  -267.06328206  -73.29909108  128.93093316]
 [   6.71011608  -62.78262301  -32.6657127   -80.00529939  256.35235515
   -63.58037876   53.02911913  -77.05757652]
 [  95.13549119 -159.33970561 -134.1771553  -267.06328206  -63.58037876
   747.14220277    1.26270451 -219.37987674]
 [  51.86167951  -96.7188796   -35.12864547  -73.29909108   53.02911913
     1.26270451  238.6557604  -139.66264741]
 [-122.58958706   75.22735047   21.47773363  128.93093316  -77.05757652
  -219.37987674 -139.66264741  333.05367046]]

In [8]: eigenvalues, eigenvectors, = np.linalg.eig(cov_matrix)
        print('Eigenvectors of Cov(X): \n%s' %eigenvectors)
        print('\nEigenvalues of Cov(X): \n%s' %eigenvalues)

Eigenvectors of Cov(X):
[[ 0.24216786 -0.23304127  0.35355339 -0.54650028  0.34900164  0.24263134
  -0.46008264 -0.26652539]
 [-0.2320557   0.21004915  0.35355339  0.28336041  0.66156373 -0.40750816
   0.14797794 -0.26666354]
 [-0.15104702 -0.0756895   0.35355339  0.19555006 -0.49779746 -0.46987952
  -0.57605919 -0.08822017]
 [-0.41563654  0.18098142  0.35355339 -0.49133856 -0.36828479  0.0496277
   0.45683638 -0.28510056]
 [ 0.05114387 -0.48868195  0.35355339 -0.17720876  0.0789403  -0.29622126
   0.28068609  0.65529232]
 [ 0.73015769  0.53226997  0.35355339  0.03110374 -0.15710617 -0.06932017
   0.14579398  0.08274941]
 [ 0.14350419 -0.47101419  0.35355339  0.52641965 -0.14266291  0.41459652
   0.25325285 -0.31472933]
 [-0.36823436  0.34512637  0.35355339  0.17861375  0.07634566  0.53607356
  -0.2484054   0.48319727]]

Eigenvalues of Cov(X):
[1.11554951e+03 4.85892625e+02 3.37428528e-12 1.22955135e+02
 2.95284929e+02 1.76427142e+02 2.32097712e+02 2.28059153e+02]
```

The *Co-Variance matrix.*



Successful Recognition Percentage and an example of a recognised face by the algorithm.

Learning Outcomes

Eigenfaces is a technique used in facial recognition, which involves extracting the principal components of a large set of training images of faces. These principal components, also known as "eigenfaces," are then used to represent the faces in a lower-dimensional feature space, which reduces the computational complexity of the recognition process.

This can provide learners with a deeper understanding of the algorithmic concepts and techniques involved in facial recognition. For instance, learners can gain insight into the mathematical operations used to calculate eigenfaces and the feature space, as well as the role of dimensionality reduction in improving recognition accuracy.

In addition, learners can also learn about the performance evaluation metrics used to measure the effectiveness of the eigenface method and the limitations of the approach. For instance, learners can understand the impact of lighting conditions, facial expressions, and other factors on the accuracy of the recognition results.

Moreover, studying the eigenface method can also provide learners with a broader perspective on the applications of computer vision and pattern recognition in various fields, such as security, biometrics, and human-computer interaction.

Overall, the knowledge gained from studying the research paper "Eigenfaces for Recognition" can serve as a foundation for further research and exploration in the field of computer vision and pattern recognition.

Methods of Improvement

There are several ways in which the eigenfaces approach to face recognition could be improved, some of which are as follows-

I. *Increase the size and diversity of the training set*: The accuracy of the system can be improved by collecting a larger set of characteristic face images of the known individuals. This set should include images with a wide range of poses, expressions, and lighting conditions. Including images of different ages, genders, and ethnicities can also improve the system's ability to recognize faces.

II. *Use more advanced algorithms for dimensionality reduction*: While the eigenfaces approach is a simple and effective method for dimensionality reduction, there are more advanced algorithms that can be used to improve accuracy. For example, non-negative matrix factorization (*NMF*) and independent component analysis (*ICA*) have been shown to outperform *PCA* in some cases.

III. *Incorporate deep learning techniques*: Deep learning has revolutionized the field of computer vision, and it has shown great promise in face recognition tasks. By training a deep neural network on a large dataset of face images, it is possible to learn highly discriminative features that can be used for recognition.

IV. *Use more efficient hardware*: We can get even faster recognition times with more advanced hardware, such as *GPUs* or custom *ASICs*.

Conclusion

The research paper "Eigenfaces for Recognition" proposed an innovative approach to face recognition using eigenfaces derived from the covariance matrix of face image data. The method demonstrated high recognition rates and robustness to facial variations, making it a significant contribution to the field of computer vision. The paper's introduction of eigenfaces, dimensionality reduction techniques, and robustness inspired further research and advancements in face recognition algorithms.

The eigenface approach has found wide application in various domains, including biometric systems, surveillance, access control, and human-computer interaction. Its simplicity, efficiency, and effectiveness have made it a popular choice for many real-world face recognition applications.

Despite these limitations, the eigenface method laid the groundwork for subsequent advancements in face recognition. Researchers have built upon the concept of eigenfaces to develop more sophisticated algorithms that address the limitations and enhance the accuracy and robustness of face recognition systems.

In conclusion, the research paper "*Eigenfaces for Recognition*" by *Turk* and *Pentland* made significant contributions to the field of face recognition. The introduction of eigenfaces and the use of *PCA* for dimensionality reduction opened up new possibilities for efficient and effective face recognition. The paper's findings and methodologies have had a profound impact on the development of subsequent algorithms and have paved the way for advancements in computer vision and biometric systems. The eigenface approach continues to be influential and remains a benchmark for evaluating and comparing face recognition techniques.

Bibliography and Tools

I. Python: 3.10.6

II. Jupyter Notebook: 6.5.4

III. Numpy: 1.24.3

IV. MatPlotLib: 3.7

V. Linux on a VM

VI. StackOverflow

VII. DataScience.StackExchange

VIII. MachineLearningMastery.com