DEDICATION

I dedicate this project to my nation, Nigeria, with a dream that one day; projects like this won’t be done just to fulfill the requirements for the accomplishment of a degree, but for proper implementation, and to aid further research.

ACKNOWLEDGEMENT

I would like to appreciate my mother, Mrs Morenike Balogun, who has been the wheel moving this cart all along, for her financial, emotional, moral, and spiritual support to the success of this program.

I'd also like to appreciate my Supervisor, Dr. S. O. Asaolu, for his support, and supervision in the most unlikely way; giving all the drive necessary to propel me to action, and still not spoon feeding me like an undergraduate student.

I could go on and on in my appreciation to all those who played a part in this one way or the other, I'd like to say, thank you all.

Most importantly, my sincere gratitude goes to God, for His endless love, and inspiration. There'd be no me without you, Lord, thank you.

ABSTRACT

As a biometric, facial recognition is a form of computer vision that uses faces to attempt to identify a person's claimed identity against a registered database.

Of all the biometrics methods, facial recognition is one of the few (if not the only) that combines authentication, surveillance, and also capable of intrusion detection without the conscious knowledge of the subject.

The main objective of this project is to design a system that takes in images of 25 faces as input, trains it using the Principal Component Analysis face recognition model. It then takes a test face, and finds a match between the new face, and the faces stored in the dataset.

The system further takes a partially covered face, and also attempt a match between the face, and faces stored in the dataset, while returning the maximum face covering for efficient recognition; beyond which the system does not recognize the test face.

CONTENTS **Page Number**

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

Biometrics is a fast growing technological means of authentication, and identity verification. It has been proven over and again to be way more secure, easier, more cost effective (in the long run) than other conventional methods of authentication. It's emergence into surveillance is rapidly increasing such that any serious minded business would be thinking of implementing one.

As a biometric, facial recognition is a form of computer vision that uses faces to attempt to identify a person's claimed identity against a registered database.

Several works have been done on facial recognition, but that doesn't necessarily rule out further research into it as people seem to believe.

1.1 HOW DOES A COMPUTER DETECT A FACE?

This might appear a trivial question given the ease with which humans distinguish a human's face from another, that of an animal, or even a design. The least intelligent human should not find it difficult to differentiate between a face and a soccer ball; but the computer does not see or think like we do. Hence, there are certain ways in which computers are trained to detect face from a cluster of figures in an image:

Skin Color: In color images, the computer can be trained to look out for the predominant color on the human body, and combine a sequence to determine if the figure is a face or not. The eye region is usually darker than the other parts of the face, the smile lines are not as dark as the eye region, but have a way of segmenting the face too, the sides of the nose have their lightening variation also, and so on. These amongst other things, are carefully studied by the computer, and once most of these descriptions are met, it is safe to declare a face.

Motion: In videos, it is common place to have the head move more than any other part of the body, and a pattern of movement or series of motion can be used in determining the face region of a human image. The movement of the lips/mouth during conversation is also a part of this motion method of detection.

Shape of the head: The shape of the head has also been used successfully to detect a face, given that the average human head has about the same oval shape, and the nose projections, the socket of the eyes, the protrusion of the ears, the stylish bulging of lips all add up to give the head a unique shape from other parts of the body.

A combination of all the above: As effective as all the above have been or can be in face detection, a more effective way to detect a human face would have to convolve all the characteristics defined above in an attempt to increase the accuracy of detection.

1.2 AIM

The aim of this project is to design a system that detects a human face from an image of a partially covered face, compares it with a database of faces, and attempts to find a match.

1.3 MOTIVATION

With the rise in technology in the Nation, expert systems seem to have a future nearer than we think and humans are at the center of it all. The need to recognize faces, perform matches is ever increasing as systems gradually replace humans in Authentication.

Presently, Nigeria is having a nationwide registration for upcoming elections, in which details are taken; which includes the face and fingerprints.

Before that, the Nigerian Communication Commission (NCC) instructed that everyone should register their SIM (also, a picture of the face was taken)[1, 2].

Registering for examinations within and outside the country, opening a Bank account, getting a driver’s license, international passport, and so many other things require a form of identification or the other. All this translates to the fact that virtually everyone would have had a registration or the other, with the picture taken, and all these can be brought to great use if a system such as this is well implemented.

The most common, and (arguably) the most important feature is the face – since it's the most easy and natural form of identification. Hence, a system that detects, and recognizes faces, will be required in almost every company and establishment before long.

1.4 BENEFITS OF THE PROJECT

With an efficient and optimal facial recognition system, the department (this research group in particular) boasts of one of the most sought after weapons of fighting crime in the Nation – among other merits.

In Examinations: Reduced cost of printing dockets and examination passes, reduced stress of signing dockets, in that; the department installs cameras which are connected to systems that authenticates students for examinations based on their registered courses. Even during the examination, checks are made (by the system) to ensure there are no impersonations.

In Medicine: High profile patients are more protected by having a system that screens people with access to the room. It can also be integrated with an alarm system, to raise an alarm in case of intrusion.

A portable device in the shape of goggles (something similar to Google glass) can be used with the system to help those suffering from Prosopagnosia (A disease that makes recognition of faces difficult, also known as face blindness), by having a database of known faces with name tags.

Banking: ATMs can have the system integrated as an alternative to using PINs, or as an additional security measure.

In law enforcement: It can be used to capture faces at a crime scene, and attempt a match with several databases for records and contact details.

Household use: The system can be setup to deny "unknown faces" access to the house (building at certain times of the day, and raise an alarm in case of intrusion).

These are just a few of its wide range of applications in the industry, and I know everyone would have a reason to implement it at one point or the other due to its wide range of applications in surveillance, access control, and human – machine interaction.

Chapter 2 contains a brief literature survey on the subject, showing previous developments on the subject, and the various algorithms and models that have been used in the achievement of working facial recognition systems. Chapter 3 explains the methodology of this thesis, showing all steps taken, and all equations used in the project. Chapter 4 shows all experiments carried out, and results achieved. Chapter 5 concludes the thesis with recommendations for future work on the subject area, and outlines other suggestions for continuous research on the subject of face recognition.

CHAPTER TWO

LITERATURE REVIEW

2.1 BACKGROUND

A literature on face recognition would probably be one of the largest to put together due to the intensity of efforts put into the technology in recent times. Research on the subject started around 1964[3], when Woodrow Wilson Bledsoe created one of the first face recognition systems; he had a volunteer, mark, import facial characteristics and calculating the difference between faces[4]. The distance were inputted into computers, and used in recognizing faces.

In 1973, the first fully functional automated system for face detection was created by Takeo Kanade[5], but was it was not produced until Kanade's Paper in 1977. After the Kanade system, there was a considerable reduction in research on the subject until around 1987, when Sirovich and Kirby did a work on a low dimensional face representation (now known as Principal Component Analysis)[6]. This sparked further researchers to re – establish their interest in the field, and in 1991, Turk and Pentland introduced the Eigenface[7], which was a big step in the field of face recognition because it reinvigorated research towards achieving better recognition algorithms from the Principal Component Analysis[8]. The Fisherface soon followed in 1996[9]. The most popular and most commonly used algorithm for face detection is the Adaboost[10] combined with the Haar cascade, which was designed by Viola and Jones in the year 2001. In 2007, Naruniec and Skarbek introduced the Gabor Jets[11].

2.2 Face recognition algorithms

Due to recent interest in biometrics as a method of authentication, verification, access control, and other merits (due to it being uneasily shared or forged), and face recognition appearing as one of the front liners in the subject, there are a number of face recognition algorithms that have been experimented on over time, some of which include:

2.2.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) was invented by Karl Pearson in 1901, as a tool in exploratory data analysis and for making predictive models – such as face recognition[12]. PCA is a mathematical algorithm that uses orthogonal transformation to convert a set of images into a set of uncorrelated variables which are also called eigenfaces. The major concept of this model is to de – correlate data in order to outline the differences and similarities in the principal components of the images being trained. It is the simplest of the true eigenvector – based multivariate analyses[13]. The PCA is often used to reveal the internal structure of a given data in a way which best depicts the major features in the data.

A set of images is visualized as a set of co – ordinates in a high – dimensional data space, the PCA then supplies the user with a lower dimensional representation.

2.2.2 Independent Component Analysis (ICA)

As opposed to the PCA which de – correlates a given set of data using the principal components of the covariance matrix of a multidimensional data, the ICA minimizes higher order dependencies in the input, with a goal of decomposing an observed signal into a linear combination of unknown independent signals[14].

Simply put, ICA is a form of generalizing the PCA, with the major difference being the higher order minimization of the input. Much of the important in a face recognition problem may be contained in higher order relationships (and not just second order relationships as represented by the PCA) of the input image pixels; hence, generalizations of the PCA are of extreme importance – with the ICA, being one of these generalization algorithms.

Several works have been done on face recognition using the Independent Component Analysis. As a result of its improved performance, there have been several algorithms for performing ICA, such as the ones proposed by Bell and Sejnowski[15], the Robust learning Algorithm for blind separation of signals by Cichocki, Unbehaun, and Rummert [16], and so on.

2.2.3 Linear Discriminant Analysis (LDA)

This model can also be called the fisher surface approach. The linear Discriminant analysis finds the linear transformation such that face feature clusters are most separable after the transformation[17]. It is one of the depth images face recognition techniques (the same class with the Principal Component Analysis, and the Independent Component Analysis)[18].

LDA performs better when used together with other face recognition models such as the PCA or the ICA. [19] showed that combining PCA and LDA improves the generalization capability of LDA when only few samples per class are available.

2.2.4 Evolutionary Pursuit (EP)

Evolutionary pursuit (EP) is a pursuit method that uses the Genetic Algorithm in processing face images in a lower dimensional whitened PCA subspace[20]. A detailed algorithm for the Evolutionary pursuit is shown in a paper written by Chengjun Liu and Harry Wechsler ([20] page 11), they give the advantages of using the EP in addition to using principal components. They further illustrate how EP expressly applies to face recognition in 2000[21].

2.2.5 Elastic Bunch Graph Matching (EBGM)

Wiskott et al showed in [22] that Elastic Bunch Graph Matching is another face recognition model, using one image per person (which is often difficult given change in poses and illumination level). The face system derives most of its recognition matrix by extracting facial landmarks in the form of graphs, which are represented by sets of jets (such as the Gabor Jets[11]).

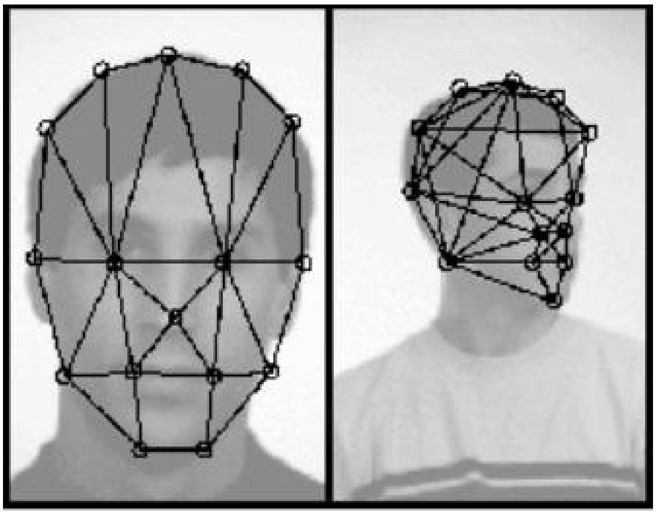

Fig 2.1: An illustration of a face graph as shown by Wiskott et al[22]

After these representations, the new image is recognized by simply comparing it with other existing image graphs.

2.2.6 Kernel Methods

The kernels methods of face recognition are improvements on the existing models by applying kernel subspace representations to face recognition. Kernel methods such as kernel PCA (kernel principal component analysis)[23], and kernel FDA (kernel Fisher discriminant analysis) have been used in the past[9], to further improve the performance of recognition algorithms.

The kernel approach works by mapping the original data into a high dimensional (or even an infinite dimensional) feature space[24]. The kernel methods are able to capture higher order statistical dependencies (just like the Independent Component Analysis), but the computations involved in the kernel methods is still maintained at about the same level as that in the non – kernel methods.

2.2.7 Three Dimensional (3 – D) Algorithms

The need for a 3 Dimensional face vector space representation was berthed in other to overcome the disadvantages and short comings of the 2D recognition models[25], especially, the challenges posed from variation in lighting conditions, varying facial expressions, different poses, and so on.

Here, there are 2 major 3 – Dimensional models described:

2.2.7.1 Three – Dimensional Morphable Model

This is a model that morphs between three dimensional objects – as used in computer graphics. This model applies the computer graphics technique to the vector space representation of face models.

The morphable face recognition technique simply follows the logic of trying to generate a face image by morphing between a training set of images given in a large database of 3D images.

The 3 – d technique of face recognition takes several mug – shots of a subject, combines them, and establishes a strong relationship between them, which can then be used to identify the face no matter the rotation or inclination of the face.

The 3D model is a significant improvement in the field of face recognition, as it takes many of the faces of a person. This method is based on the illumination cone to deal with illumination variation.

2.2.7.2 Limitations of the 3D recognition models

Generally, 3 dimensional face recognition models seem to have a high success rate, given the quality and quantity of information gathered about the faces in the training set. Several views of the face are considered, and registered – which gives a better overall view of the face, such as the one seen by the human eye – but in practice, it is not always possible (due to time constraints, ignorance, cost implication, and other obstacles) to get several image shots of the registered face. Hence, the 3D models are more used in laboratories for research purposes, and in law enforcement agencies – such as the Police Force, where mug – shots of suspects and convicted criminals are taken regularly for detailed recognition.

2.2.8 Bayesian Framework

The Bayesian framework assumes that we always have a prior distribution for everything; that is, for any event in question, there must be a probability that the event will happen[26]. The prior might be vague, but when we see some data, we combine our prior distribution with a likelihood term to get a posterior distribution. The likelihood term takes into account how probable the observed data is given the parameters of the model[27].

[27] compared the Bayesian framework to the eigenface matching, and achieved a reasonable success in recognizing plain faces – without disguises – when parsed through the system, but had no significant improvement on the PCA with which it was compared; and in fact, the PCA performed better on larger datasets than the Bayesian framework.

2.2.9 Support Vector Machine (SVM)

The Support Vector Machine (SVM) is one of a range of classifiers under the supervised machine learning algorithm. Supervised learning is a learning algorithm that determines the best fitting model that correctly maps a set of inputs to outputs. It comprises of a number of known combinations (input and output), but tries to predict the output given a training function; errors are corrected, and the system is retrained until it achieves a reasonable amount of success with which it can then be tested with new input, to determine the output.

The SVM is a pattern recognition tool with binary classifiers that attempt to find the hyper – plane that maximizes the margin between the positive and negative observations for a fixed class[28].

2.2.10 Hidden Markov Models (HMM)

The Hidden Markov Models have a discrete one – of – N hidden state. Transitions between states are stochastic and controlled by a transition matrix. The outputs produced by a state are also stochastic. We cannot be sure which state produced a given output. So the state is “Hidden”. It is easy to represent a probability distribution across N states with N numbers. To predict the next output, we would need to infer the probability distribution over hidden states. Hidden Markov Models have efficient algorithms for inference and learning, and that is what makes them appropriate for speech – in the 70s, they took over speech recognition applications.

2.2.10.1 Limitations of the HMM

At each step, it must select one of its hidden states, so, with N hidden states, it can only remember log(N) bits about what it generated so far.

For example, in speech recognition application for the HMM, the syntax needs to fit, the semantics, intonation, accent, rate, volume, and vocal tract characteristics all must fit – which is not always possible for real life problems.

2.2.11 Active Appearance Model (AAM)

This is a model with a unique application in facial recognition, as it recognizes facial actions – or facial expression recognition as often called – as its name implies. It has been used successfully in identifying the differences between genuine and – genuine pain, detecting people lying by observing their facial expression, amongst other applications.

The AAM is one of a number of recognition models in the Facial Action Coding System (FACS) family. It uses pattern recognition like many other algorithms, given a face image; it takes the pixel values – of basic alignments like the eyes, and nose – and derives an effective representation from it, in other to draw a pattern. This pattern is examined amongst a database of such stored expressions, and a result is returned.

The models listed above are just a few of the models employed in face recognition (to say the least); other methods of pattern recognition, and machine learning are being explored in a search for the true and accurate recognition system [16, 18, 29-33]. Aside using these methods individually, several individuals have tried combining some of these models in an attempt to enhance the detection, and recognition of a face [14, 19, 34], while some have simply taken to comparing models by presenting the same setup to each model, and observing the results, advantages, and short comings of each model [25, 34-36].

The most recent development in the area of face detection is the deep dense face detector designed by Farfade et al, which was published on the 10th February, 2015[37]. The system is capable of detecting rotated, distant, and even partially occluded faces with a high degree of accuracy. It uses the deep learning architecture known as deep convolutional neural network.

The system was trained with a dataset of 200,000 positive and 20 million negative training examples. For fine tuning, 50,000 iterations were used, with a batch size of 128 images – where each batch contained 32 positive, and 96 negative examples. These features were parsed into an 8 – layer deep convolutional neural network, where its first 5 layers were convolutional, and its last 3 layers were fully connected.

CHAPTER 3

METHODOLOGY

The stages involved in the building of this system are:

3.1 **Face detection**

This is the first stage of the project, in which the system is trained to detect a face or faces in a given image. As highlighted in the introduction, face detection is a seemingly easy, but complicated, and important part of the face recognition system. The method used in this project is the Viola – Jones object detection framework based on Haar Cascades.

To put it simply, Paul Viola and Michael Jones wrote an algorithm that could spot faces in an image in real time. They noticed that the bridge of the nose forms a brighter region than the eye sockets near it, and that the eyes formed a dark horizontal region. So, they built an algorithm that looks for vertical bright regions that might be nose, and horizontal dark regions that might be eyes, and also looks for other general patterns synonymous with a face.

When these features are observed individually, they do not convince to be a face; but when they are detected in a cascade, they perform better, and represent one of the fastest face detection algorithms. A Haar cascade is a series of Haar – like features that combine to form a classifier. A Haar – like feature is a rectangular pattern in data. It simply examines certain areas of an image, comparing how they vary from one another in brightness.

The Haar cascade takes a feature scale, and slides a rectangular block across it. The average pixel values under the bright area and the dark area are computed, if the difference between the areas is above some threshold, the feature is considered to match. This is used to detect the presence of a feature by comparing the brightness and darkness of certain areas of a face.

There are three basic set of Haar – like features used in classification:

- Edge features

- Line features
- Center – surround features

Using a single classifier isn't accurate enough, it's sometimes called a weak classifier; hence, Haar cascades consists of a number of weak classifiers combined in an order that increases the accuracy of the detection.

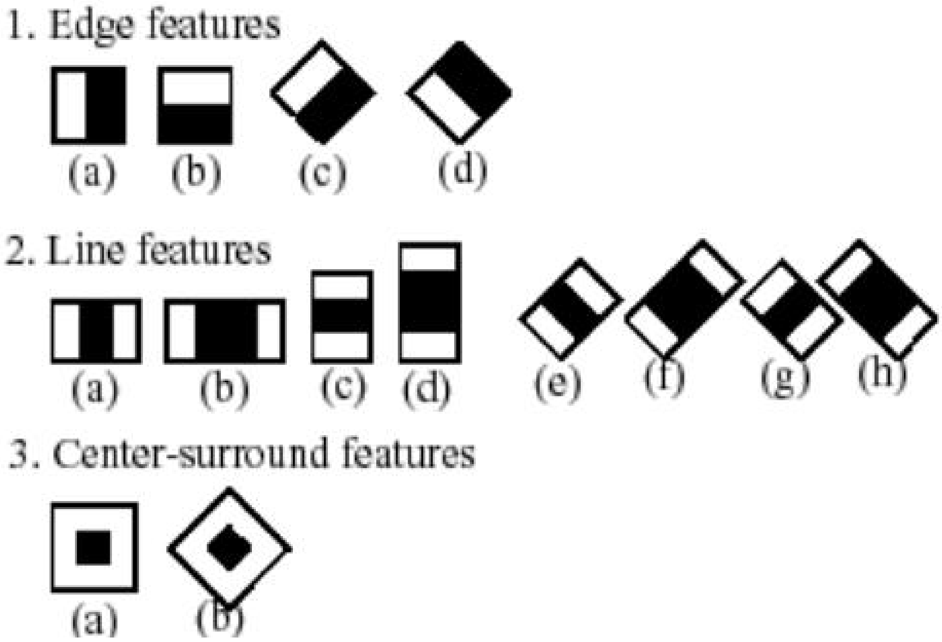

Fig 3.1: A pictorial example of Haar cascades combination

Adaboost tries out multiple weak classifiers over several rounds, selecting the best weak classifier in each round, and combining the best weak classifiers to create a strong classifier. It's a form of supervised learning in which the system is presented with a known result – a face – and trained to confirm the face based on its classifiers, such that faces on new images can be detected.

3.2 **Feature Extraction**:

The Principal Component Analysis used for in this project, uses a holistic matching approach for face recognition – which is a form of classification using the whole face region.

The features extracted from each face are stored as a dataset, with identifiers denoting what image has what features.

Alternatively, if the system employed one of the feature based techniques, some of these feature extraction algorithms would have been used:

Harris corner detection: This is a corner detection algorithm, which finds the corners in a grayscale image. It returns corner locations as a matrix of [x, y] coordinates. It is a corner detection model which works in a way similar to the minimum eigenvalue by Shi and Tomasi, or local intensity comparison method such as FAST algorithm by Rosten and Drummond.

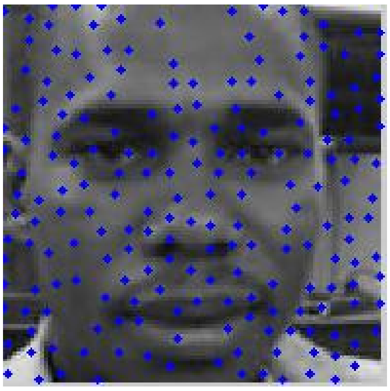

Fig 3.2: An example of feature extraction using the Harris Corner detector

MSER (Maximally stable external regions): This is an algorithm that returns MSER regions containing information about MSER features detected in the 2 – D grayscale input image.

Fig 3.3: An example of feature extraction using the MSER algorithm

SURF (Speeded up robust features): This is a feature algorithm that returns SURFPoints object known as blob features, containing information about SURF features detected in the 2 – D grayscale image.

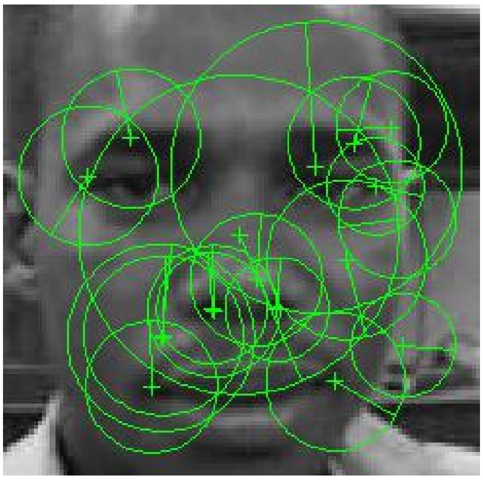

Fig 3.4: An example of feature extraction using the SURF algorithm

3.3 **Face recognition**

At this stage, the output of the matching is given; if a face is recognized as being in the database, the system returns the identity of the stored version of the recognized face; else, the system returns an error message that face is not found in the database.

There are several proven methods for doing this – as stated in the literature review – but involving learning using the newly developed deep learning architectures have shown a higher level of accuracy in the recognition of faces than other algorithms[37]. This stage uses the Principal Component Analysis in the recognition of the face.

The Principal Component Analysis – also known as Eigenface – is a recognition algorithm that begins with a database of faces to be used for the training of the system.

The faces are converted into grayscale, from which the average face is created – which is the common features between the images – and subtracted from each individual face in the database, to obtain a database of new faces, known as the unique features in each face (what makes the faces differ from others).

Each image (new faces) is then formed into a column vector of its pixels, and all vectors are brought together to form another matrix. The covariance of the matrix is then calculated.

Once we have the co – variance, we calculate the eigenvectors, and eigenvalues of the matrix.

The eigenvalues are then multiplied with the faces in the database (that's with the average face subtracted). All elements are added together to form the eigenfaces.

A test face is then taken, multiplied by a transpose of the eigenfaces, and the average face is subtracted from the result, to obtain a weight. The Euclidean distance between the weight faces are then calculated; the face with the least distance is considered the matched face.

It would be adviseable to set a threshold, such that a value below the threshold is determined as a match, and values above the threshold are either considered a non – match, or not being a face at all.

3.4 **Partial occlusion and recognition**

Since open face detection and recognition has been achieved, the project moves further to the stage of partial occlusion of the face region, with different sizes of masks placed on the face region, to test the limit of covering the system allows for proper recognition to be possible.

CHAPTER 4

EXPERIMENTS AND RESULTS

This project uses a dataset of 68 images, consisting of a training set of 25 single face images, and a test set of 43 images of people with system generated partial occlusions was used in this project, consisting of both male and female of the working age range. The images were taken under different lighting conditions, with different poses, and different facial expressions and resized to 110 x 110 pixels after the faces had been detected using the Viola – Jones face detector.

The purpose of altering the lighting conditions, poses, and facial expression, is to observe the effect of the irregularities in the training process – as most projects done on face recognition use balanced, even lighting conditions, poses, and facial expressions for training; though the test set may have some variations – and show how it would perform on the test sets.

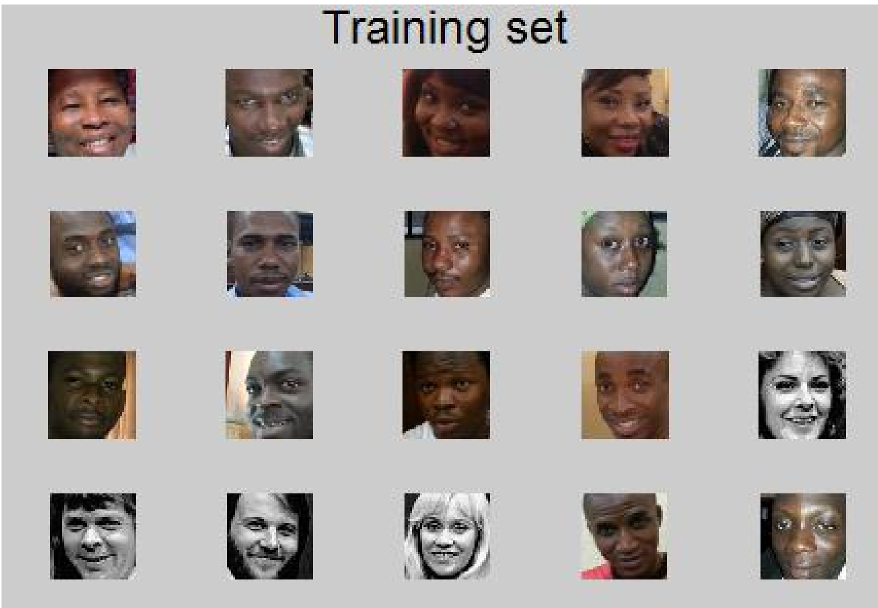

Fig 4.1: Training set

The images were then normalized to balance the illumination, and achieve a better even image set.

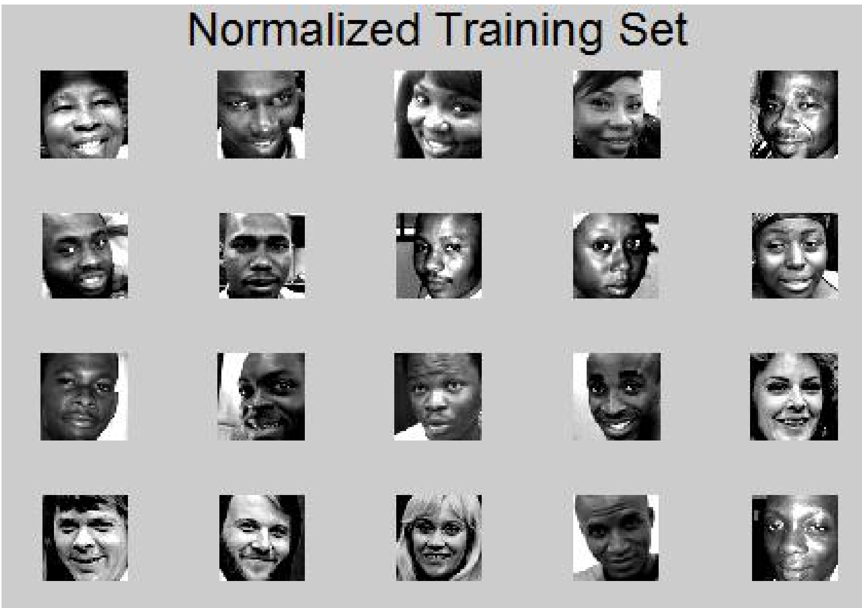

Fig 4.2: Normalized image set

The mean image was then obtained by combining all the images to deduce the common features between them.

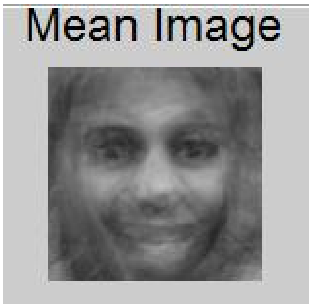

Fig 4.3: Average face image

With the mean image obtained, the faces left are distinct features of each face (that is, after subtracting the mean image from all faces in the dataset).

The eigenfaces were calculated, and presented to the system for training.

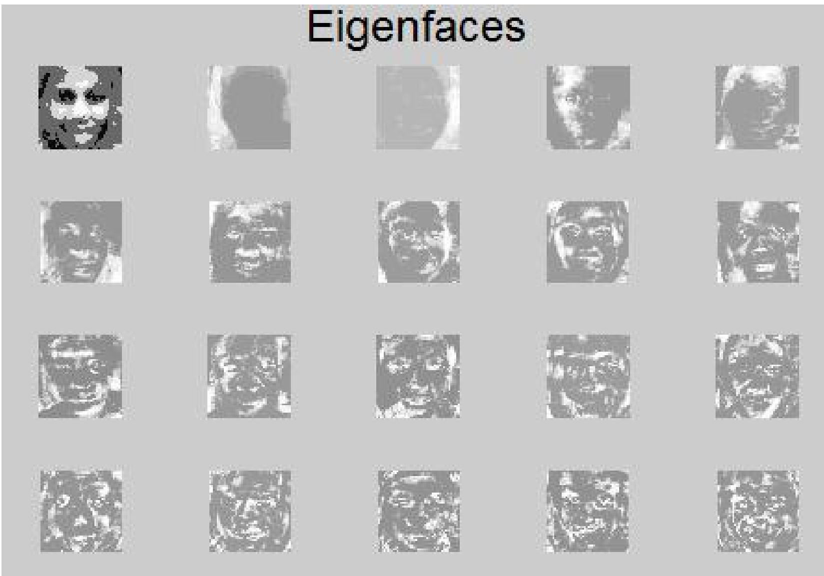

Fig 4.4: Eigen faces of the training images

In line with the PCA model, each face image in the dataset is assigned a weight, with which it is identified differently from every other face in the training set.

A test image is then entered to the system, the face is normalized, and the weight of the face is compared with other faces in the set by calculating the Euclidean distance between them.

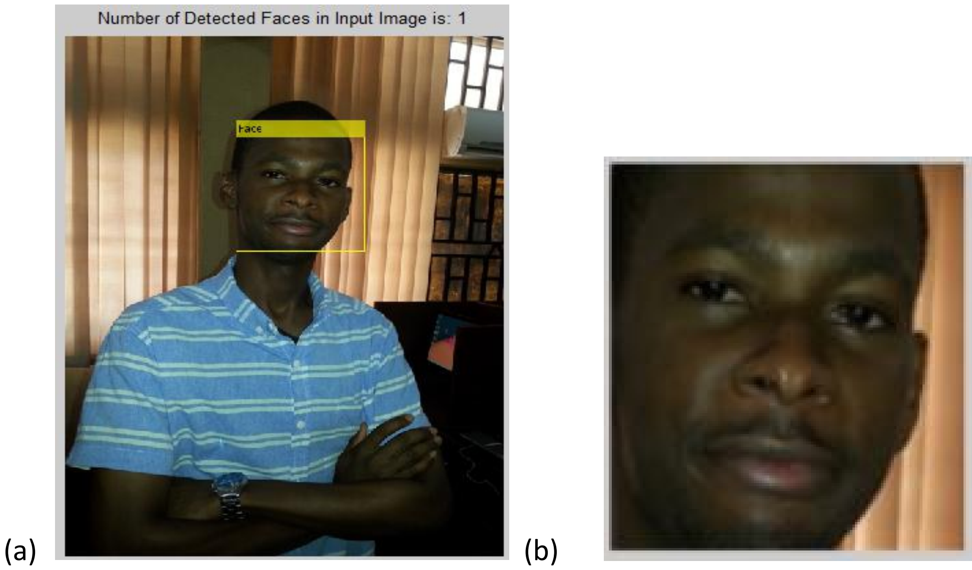

Fig 4.5: Test image bring pre – processed: (a) Face detection from the original image (b) Face region cropped from entire image

The image with the lowest distance from the input image is deemed the closest, and therefore, is returned as the recognized face in the system.

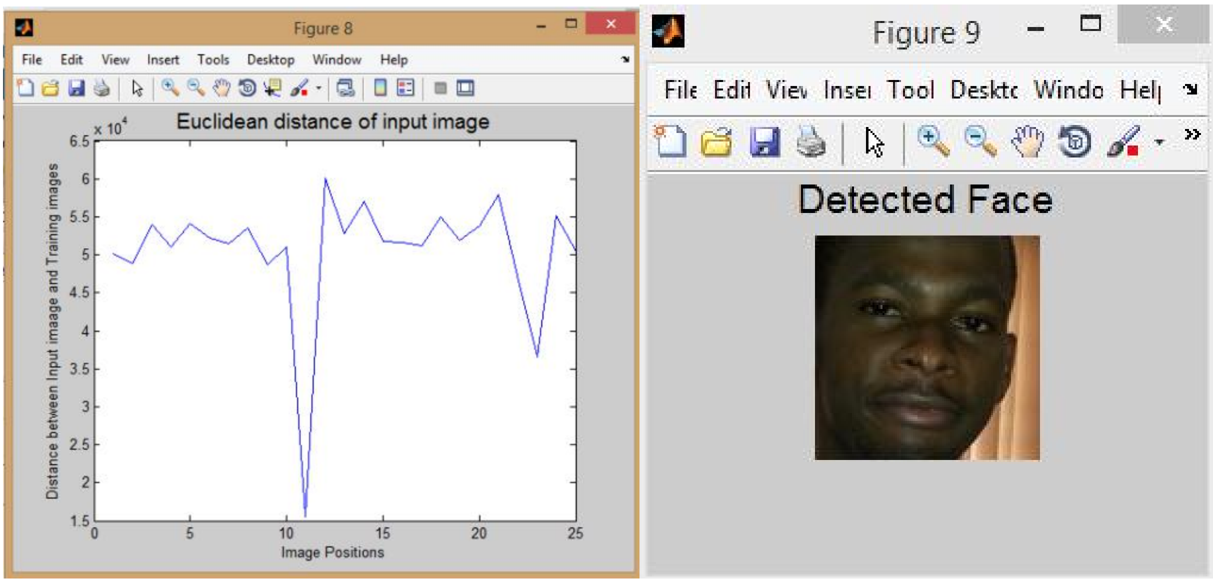

Fig 4.6: The Euclidean graph of the input image with the training images (left), and the system detected face (right).

Since the test image is uncovered, the distance between the correctly detected face is lower than other faces by a reasonable margin, showing that the system is able to easily detect a face image without occlusion.

Next, a new test image is entered, but this time, with a partial occlusion (NB: this project employs electronically induced occlusions). The system does the same thing as above, and also draws a reconstructed face image, to eliminate the occlusion, while matching other – exposed – parts of the face with other faces in the set, and also attempt a recognition using the Euclidean distance.

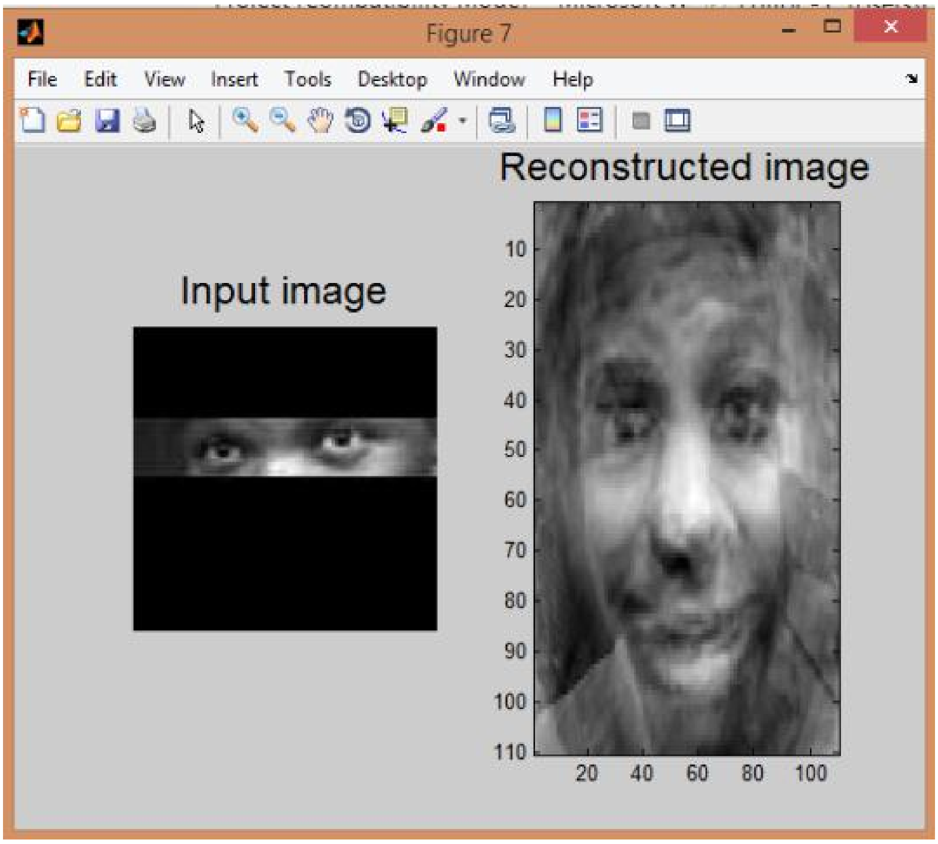

Fig 4.7: Occluded face image and the reconstructed image at 80% occlusion

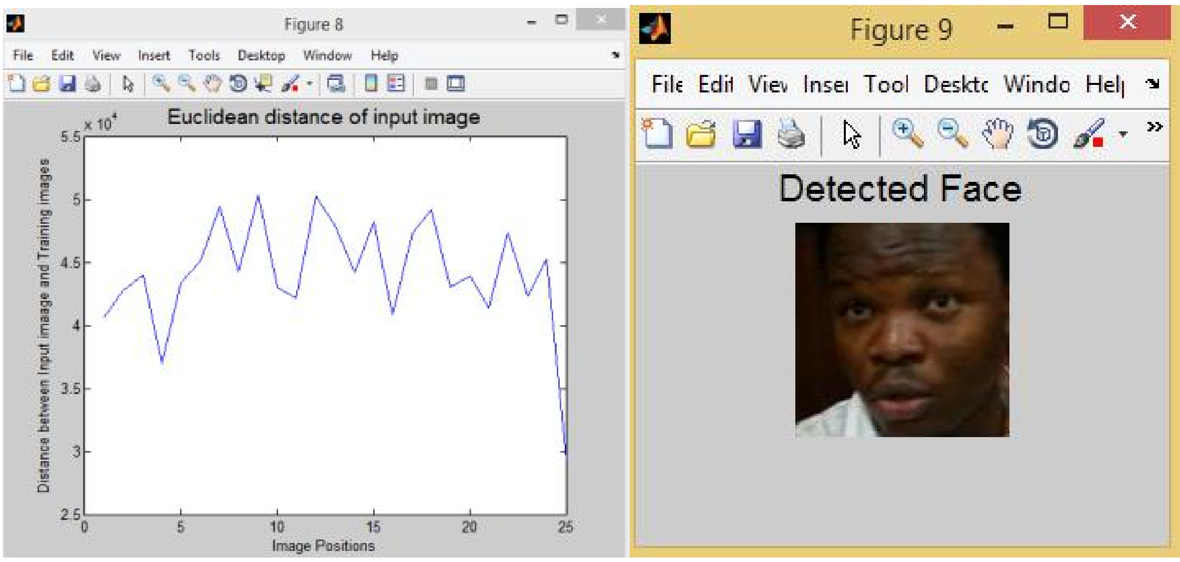

Fig 4.8: Euclidean graph of the test image with other images (left), and the system detected face image (right)

It can be seen from the second test image (with the occlusion), that the system tries to reconstruct the occluded part of the face image – using the mean image stored from the training set – and then uses the remaining (exposed) part of the image to attempt a match. The result achieved is seen from the Euclidean graph, as being the exact face, but the distance is now closer to the other faces than the set without occlusion.

Further experiments using different percentage of occlusion showed varying results:

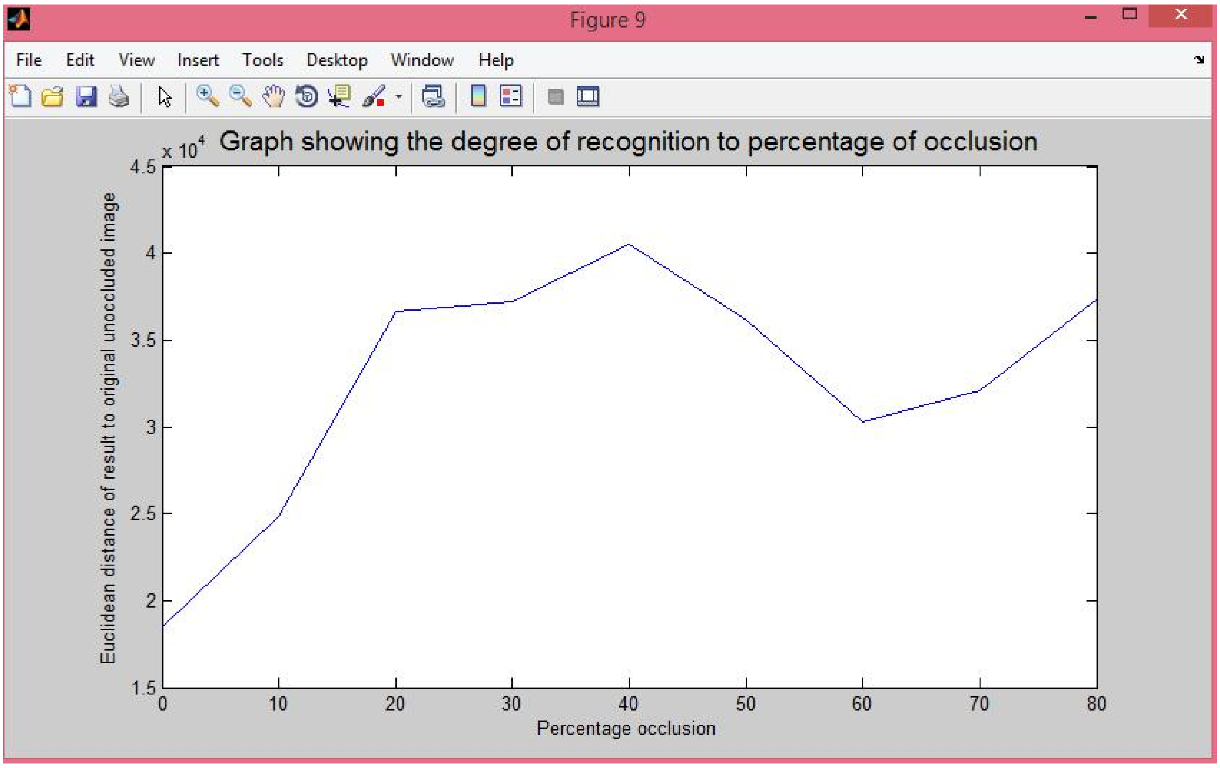

Fig 4.9: Graph showing the degree of recognition to percentage of occlusion

From the graph above, it can be seen that the system finds the recognition difficult as the occlusion increases, but there is an observation of note just after 40 percent occlusion; the recognition becomes easier as until 60 percent where the system experiences difficulties again. This shows that there are certain regions of the face that are more distinct than others, especially around the eyes – though this is also subject to the pose, and facial expression of the individual – which is evident from the figure below:

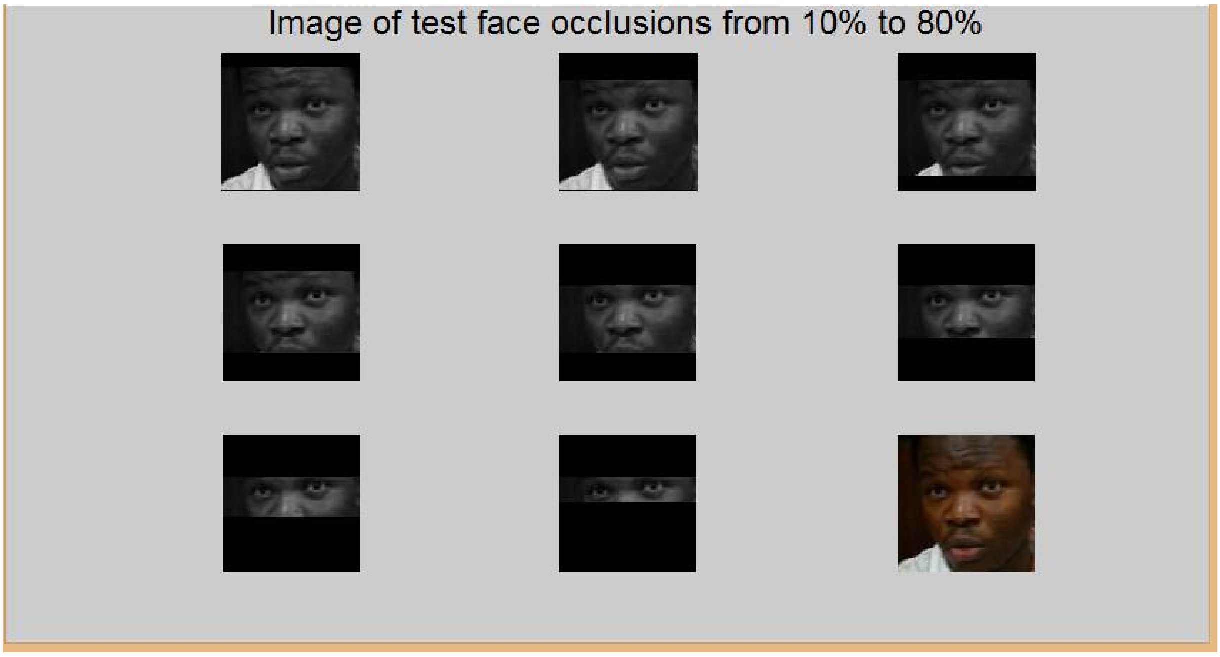

Fig 4.10: Image set of test occlusion faces used

It will be seen from the image set above and the graph, that the occlusion around the 50 and 60 percent occlusion leaves a face region similar to the average face of a matured male; hence, the decrease in accuracy.

Accuracy

The accuracy of the system (which is the degree of precision in the recognition of faces), is given by the following:

$$\text{Accuracy (\%)} = 100\% - \frac{FAR\% + FRR\%}{2}$$

Where FAR and FRR are False Acceptance rate and False Rejection rate respectively.

The false acceptance rate is calculated as:

$$\text{FAR} = \frac{Wrongly\ matched\ individuals}{Total\ number\ of\ faces\ not\ in\ Training\ set}$$

The false rejection rate is calculated as:

$$\text{FRR} = \frac{Wrongly\ rejected\ faces\ that\ are\ in\ training\ set}{Total\ number\ of\ faces\ in\ training\ set}$$

But since the scope of this project is of particular concern to the occlusion, this part of the project will be left for further research as it entails analyzing every image in the dataset, testing each for false rejection and acceptance, and as well doing the same for the test set.

The accuracy of the partially occluded images can be seen to be 40% although a face image with an occlusion of up to 80% was still successfully recognized, but this wasn't the result for all faces.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

From the experiments carried out, it has been shown that recognition of a partially occluded face – up to 80 percent occlusion – is achievable with confidence scores using Principal Component Analysis.

For future research, I would suggest using other machine learning techniques – especially the deep learning architectures – as the machine learning makes pattern recognition easier, and makes it more natural, with more success recorded than when using conventional algorithms. The system would take a dataset of various occluded images, with the known identity, and is trained over any of the proven architectures such as convolutional neural network, or the deep belief network.

It would also be recommended that subsequent research work in this field be done with more images with various poses, rotations, illumination conditions, and several partial occlusions in the training phase as this helps in perfectly training the system to understand what various poses and illuminations are possible.

REFERENCES

[1] C. E. Izuogu, "Data protection and other implications in the ongoing SIM card registration process," *Available at SSRN 1597665,* 2010.

[2] N. Jentzsch, "Implications of mandatory registration of mobile phone users in Africa," *Telecommunications Policy,* vol. 36, pp. 608-620, 2012.

[3] A. O. Boyer and R. S. Boyer, "A biographical sketch of WW Bledsoe," in *Automated Reasoning*, ed: Springer, 1991, pp. 1-29.

[4] A. L. George and J. L. George, *Woodrow Wilson and Colonel House: a personality study* vol. 1144: Courier Corporation, 1964.

[5] T. Kanade, "Picture processing system by computer complex and recognition of human faces," 1974.

[6] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *JOSA A,* vol. 4, pp. 519-524, 1987.

[7] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience,* vol. 3, pp. 71-86, 1991.

[8] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, 1991, pp. 586-591.

[9] B. Scholkopft and K.-R. Mullert, "Fisher discriminant analysis with kernels," in *Proceedings of the 1999 IEEE Signal Processing Society Workshop Neural Networks for Signal Processing IX, Madison, WI, USA*, 1999, pp. 23-25.

[10] P. Viola and M. Jones, "Fast and robust classification using asymmetric adaboost and a detector cascade," *Advances in Neural Information Processing System,* vol. 14, 2001.

[11] O. Ayinde and Y.-H. Yang, "Face recognition approach based on rank correlation of Gabor-filtered images," *Pattern Recognition,* vol. 35, pp. 1275-1289, 2002.

[12] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *Acm Computing Surveys (CSUR),* vol. 35, pp. 399-458, 2003.

[13] J. Yang, D. Zhang, A. F. Frangi, and J.-y. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 26, pp. 131-137, 2004.

[14] S. Choi, A. Cichocki, H.-M. Park, and S.-Y. Lee, "Blind source separation and independent component analysis: A review," *Neural Information Processing-Letters and Reviews,* vol. 6, pp. 1-57, 2005.

[15] A. J. Bell and T. J. Sejnowski, "The “independent components” of natural scenes are edge filters," *Vision research,* vol. 37, pp. 3327-3338, 1997.

[16] A. Cichocki, R. Unbehauen, L. Moszczynski, and E. Rummert, "A new on-line adaptive learning algorithm for blind separation of source signals," in *Proc. ISANN*, 1994, pp. 406-411.

[17] F. Z. Chelali, A. Djeradi, and R. Djeradi, "Linear discriminant analysis for face recognition," in *Multimedia Computing and Systems, 2009. ICMCS'09. International Conference on*, 2009, pp. 1-10.

[18] G. G. Gordon, "Face recognition based on depth and curvature features," in *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, 1992, pp. 808-810.

[19] W. Zhao, A. Krishnaswamy, R. Chellappa, D. L. Swets, and J. Weng, "Discriminant analysis of principal components for face recognition," in *Face Recognition*, ed: Springer, 1998, pp. 73-85.

[20] C. Liu and H. Wechsler, "Face recognition using evolutionary pursuit," in *Computer Vision—ECCV’98*, ed: Springer, 1998, pp. 596-612.

[21] C. Liu and H. Wechsler, "Evolutionary pursuit and its application to face recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 22, pp. 570-582, 2000.
[22] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. Von Der Malsburg, "Face recognition by elastic bunch graph matching," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 19, pp. 775-779, 1997.
[23] M.-H. Yang, N. Ahuja, and D. Kriegman, "Face recognition using kernel eigenfaces," in *Image processing, 2000. proceedings. 2000 international conference on*, 2000, pp. 37-40.
[24] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Face recognition using kernel direct discriminant analysis algorithms," *Neural Networks, IEEE Transactions on,* vol. 14, pp. 117-126, 2003.
[25] A. F. Abate, M. Nappi, D. Riccio, and G. Sabatino, "2D and 3D face recognition: A survey," *Pattern Recognition Letters,* vol. 28, pp. 1885-1906, 2007.
[26] S. K. Andersen, K. G. Olesen, F. V. Jensen, and F. Jensen, "HUGIN-A Shell for Building Bayesian Belief Universes for Expert Systems," in *IJCAI*, 1989, pp. 1080-1085.
[27] B. Moghaddam, T. Jebara, and A. Pentland, "Bayesian face recognition," *Pattern Recognition,* vol. 33, pp. 1771-1782, 2000.
[28] S. Lucey, A. B. Ashraf, and J. F. Cohn, *Investigating spontaneous facial action recognition through aam representations of the face*: INTECH Open Access Publisher, 2007.
[29] M. J. Er, S. Wu, J. Lu, and H. L. Toh, "Face recognition with radial basis function (RBF) neural networks," *Neural Networks, IEEE Transactions on,* vol. 13, pp. 697-710, 2002.
[30] Y. Gao and M. K. Leung, "Face recognition using line edge map," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 24, pp. 764-779, 2002.
[31] S.-H. Lin, S.-Y. Kung, and L.-J. Lin, "Face recognition/detection by probabilistic decision-based neural network," *Neural Networks, IEEE Transactions on,* vol. 8, pp. 114-132, 1997.
[32] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 31, pp. 210-227, 2009.
[33] S. Srisuk, M. Petrou, W. Kurutach, and A. Kadyrov, "Face authentication using the trace transform," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2003, pp. I-305-I-312 vol. 1.
[34] J. Zhang, Y. Yan, and M. Lades, "Face recognition: eigenface, elastic matching, and neural nets," *Proceedings of the IEEE,* vol. 85, pp. 1423-1435, 1997.
[35] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE transactions on pattern analysis and machine intelligence,* vol. 15, pp. 1042-1052, 1993.
[36] M.-H. Yang, "Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods," in *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2002, pp. 0215-0215.
[37] S. S. Farfade, M. Saberian, and L.-J. Li, "Multi-view Face Detection Using Deep Convolutional Neural Networks," *arXiv preprint arXiv:1502.02766,* 2015.

APPENDIX A

An image can be represented as an N X N matrix A – by converting it to a grayscale image (eliminating the 3 dimensionality in the RGB image), and is said to have an eigenvector V, and corresponding eigenvalue λ if

$$AV = \lambda V \qquad (A.1)$$

The equation above (A.1) can hold if

$$\det|A - \lambda I| = 0 \qquad (A.2)$$

Further expansion will show that equation A.2 is an Nth degree polynomial in λ, whose roots are the eigenvalues.

A matrix is called symmetric if it is equal to its transpose,

$$A = A^T \text{ or } a_{ij} = a_{ji} \qquad (A.3)$$

it is termed orthogonal if its transpose equals its inverse,

$$A^T A = A A^T = I \qquad (A.4)$$

I = an N X N identity matrix

finally, a real matrix is called normal if it commutes with is transpose,

$$A A^T = A^T A \qquad (A.5)$$

Theorem: Eigenvalues of a real symmetric matrix are all real. Contrariwise, the eigenvalues of a real non – symmetric matrix may include real values, but may also include pairs of complex conjugate values. The eigenvalues of a normal matrix with non – degenerate eigenvalues are complete and orthogonal, spanning the N – dimensional vector space.

After giving some insight on the terms that are going to be used in the evaluation of the eigenfaces, we can deal with the actual process of finding these eigenfaces.

Let the training set of face images be $\Gamma_1, \Gamma_2, \ldots, \Gamma_M$ then the average of the set is defined by

$$\Psi = \frac{1}{M}\sum_{n=1}^{M} \Gamma_n \qquad (A.6)$$

Each face differs from the average by the vector

$$\Phi_i = \Gamma_i - \Psi \qquad (A.7)$$

Where Φ is the normalized face image. An example training set is shown in Figure 4.2, with the average face (Ψ) shown in figure 4.3.

This set of very large vectors is then subject to principal component analysis, which seeks a set of M orthonormal vectors, U_n, which best describes the distribution of the data. The kth vector, U_k, is chosen such that

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M} (U_k^T \Phi_n)^2 \qquad (A.8)$$

is a maximum, subject to

$$U_l^T U_k = \delta_{lk} = \begin{cases} l, \; if \; l = k \\ 0, \; otherwise \end{cases} \qquad (A.9)$$

The vectors U_k and scalars λ_k are the eigenvectors and eigenvalues, respectively of the covariance matrix

$$C = \frac{1}{M}\sum_{n=1}^{M} \Phi_n \Phi_n^T = AA^T \qquad (A.10)$$

Where the matrix A = [Φ_1 Φ_2… Φ_M] the covariance matrix C, however is N^2 X N^2 real symmetric matrix, and determining the N^2 eigenvectors and eigenvalues is an intractable task for typical image sizes. We need a computationally feasible method to find these eigenvectors.

If the number of data points in the image space is less than the dimension of the space ($M < N^2$), there will be only M-1, rather than N^2, meaningful eigenvectors. The remaining eigenvectors will have associated eigenvalues of zero. We can solve for the N2 dimensional eigenvectors in this case by first solving the eigenvectors of an M X M matrix such as solving 20 X 20 matrix rather than a 12100 X 12100 matrix and then, taking appropriate linear combinations of the face images Φ_i.

Consider the eigenvectors V_i of A^TA such that

$$A^TAV_i = \mu_i V_i \tag{A.11}$$

Premultiplying both sides by A, we have

$$AA^T AV_i = \mu_i AV_i \tag{A.12}$$

From which we see that AV_i are the eigenvectors of $C = AA^T$

Following these analysis, we construct the M X M matrix $L = A^TA$,

Where $L_{mn} = \Phi_m^T \Phi_n$, and find the M eigenvectors, V_i, of L.

These vectors determine linear combinations of the M training set face images to form the eigenfaces U_i.

$$U_l = \sum_{k=1}^{M} V_{lk}\Phi_k \tag{A.13}$$

$$l = 1, \ldots, M$$

With this analysis, the calculations are greatly reduced, from the order of the number of pixels in the images (N^2) to the order of the number of images in the training set (M). in practice, the training set of face images will be relatively small ($M << N^2$), and the calculations become quite manageable. The associated eigenvalues allow us to rank the eigenvectors according to their usefulness in characterizing the variation among the images.

APPENDIX B

```
%% Project title section
%This program is written by Balogun Emmanuel Oloruntobi....
%It is the design of a face recognition system with partial
occlusion....
%using Principal Component Analysis.
% All or part of the code may be used for research or academic
purposes...
% only, with a permission from the Author
% E - mail: beoloruntobi@yahoo.co.uk
%% Clear system memory of all previously used variable
assignments
clear;
close all;
clc
%% load all pictures into a variable pics
pics = dir('*.jpg');
% number of images on your training set.
M = numel(pics);

%Chosen std and mean.
%It can be any number that it is close to the std and mean of
most of the images.
um=100;
ustd=90;

%read and show images(bmp);
S = [];    %img matrix
figure(1);

%% Corner Detection
for i=1:M
    str=strcat(int2str(i),'.jpg');   %concatenates two strings
that form the name of the image
    eval('img=imread(str);');
    subplot(ceil(sqrt(M)),ceil(sqrt(M)),i)
    imshow(img)
    if i==ceil((sqrt(M))/2)
        title('Training set','fontsize',18)
    end
    drawnow;
   img = rgb2gray(img);
    %cornerDetector = vision.CornerDetector('Method', 'Local
intensity comparison (Rosten & Drummond)');
    %cornerDetector.MetricMatrixOutputPort = true;
    %[loc metric] = step(cornerDetector, img);
```

```
    %img = metric;
    [irow, icol]=size(img);    % get the number of rows (N1) and
columns (N2)
    temp=reshape(img',irow*icol,1);     %creates a (N1*N2)x1
matrix

    S =[S temp];          %X is a N1*N2xM matrix after finishing
the sequence
                         %this is our S
end
%% Displaying
fprintf('Program paused. Press enter to continue.\n');
pause;
%Here we change the mean and std of all images. We normalize all
images.
%This is done to reduce the error due to lighting conditions.
for i=1:size(S,2)
    temp=double(S(:,i));
    m = mean(temp);
    st = std(temp);
    S(:,i)=(temp-m)*ustd/st+um;
end

%show normalized images
figure(2);


for i=1:M
    str=strcat(int2str(i),'.jpg');
    img=reshape(S(:,i),icol,irow);
    img=img';
   % eval('imwrite(img,str)');
    subplot(ceil(sqrt(M)),ceil(sqrt(M)),i)
    imshow(img)
    drawnow;
    if i==ceil((sqrt(M))/2)
        title('Normalized Training Set','fontsize',18)
    end
end

fprintf('Program paused. Press enter to continue.\n');
pause;

%% mean image;
m=mean(S,2);   %obtains the mean of each row instead of each
column
```

```
tmimg=uint8(m);   %converts to unsigned 8-bit integer. Values
range from 0 to 255
img=reshape(tmimg,icol,irow);    %takes the N1*N2x1 vector and
creates a N2xN1 matrix
img=img';       %creates a N1xN2 matrix by transposing the image.
figure(3);
imshow(img);
title('Mean Image','fontsize',18)

fprintf('Program paused. Press enter to continue.\n');
pause;

% Change image for manipulation
dbx=[];   % A matrix
for i=1:M
    temp=double(S(:,i));
    dbx=[dbx temp];
end

%Covariance matrix C = A'A
C = dbx' * dbx;
%C = C + 2 * eye(size(C));
% Cvector are the eigenvector for C
% Cvalue are the eigenvalue C
[Cvector, Cvalue]=eig(C);
% Sort and eliminate those whose eigenvalue is zero
v = [];
d = [];
for i = 1:size(Cvector,2)
    if(Cvalue(i,i)>1e-3)
        v(:, i) = Cvector(:,i);
        d = diag(Cvalue);
    end
 end

 %sort,  will return an ascending sequence
 [B, index] = sort(d);
 ind = zeros(size(index));
 dtemp = zeros(size(index));
 vtemp = zeros(size(v));
 len = length(index);
 for i = 1:len
    dtemp(i) = B(len+1-i);
    ind(i) = len+1-index(i);
    vtemp(:,ind(i)) = v(:,i);
 end
 d=dtemp;
```

```
 v=vtemp;

%Normalization of eigenvectors
 for i=1:size(v,2)       %access each column
   k=v(:,i);
   temp=sqrt(sum(k.^2));
   v(:,i)=v(:,i)./temp;
end

%Normalization of Eigenvalues of C matrix
u=[];
for i=1:size(v,2)
    temp=sqrt(d(i));
    u=[u (dbx*v(:,i))./temp];
end

for i=1:size(u,2)
   kk =u(:,i);
   temp=sqrt(sum(kk.^2));
    u(:,i)=u(:,i)./temp;
end


% show eigenfaces;
figure(4);



for i=1:size(u,2)
    img=reshape(u(:,i),icol,irow);
    img=img';
    img=histeq(img,255);
    subplot(ceil(sqrt(M)),ceil(sqrt(M)),i)
    imshow(img)
    drawnow;
    if i==ceil((sqrt(M))/2)
        title('Eigenfaces','fontsize',18)
    end
end

fprintf('Program paused. Press enter to continue.\n');
pause;
% Find the weight of each face in the training set.
omega = [];
for h=1:size(dbx,2)
    WW=[];
```

```
    for i=1:size(u,2)
        t = u(:,i)';
        WeightOfImage = dot(t,dbx(:,h)');
        WW = [WW; WeightOfImage];
    end
    omega = [omega WW];
end


%% Ask user to input the face to be recognized
InputImage = input('Please enter the name of the image and its
extension \n','s');
InputImage = imread(strcat('C:\Users\Olorunsegun
Balogun\Desktop\Pics\',InputImage));
%% Cropping the face from the input image
FDetect = vision.CascadeObjectDetector;
FDetect.ScaleFactor = 1.025;
%Returns Bounding Box values based on number of objects
BB = step(FDetect,InputImage);
n = size(BB, 1);
if n > 0
rect = BB;
for i = 1:n
rect(i, :) = BB(i, :);
str = strcat('face ', int2str(i),'.png');
    uni = rect(i, :);
    unis = imcrop(InputImage, uni);
    unis = imresize(unis, [110, 110]);
    eval('imwrite(unis,str)');
    figure(5)
    subplot(ceil(sqrt(n)),ceil(sqrt(n)),i)
    imshow(unis)
    drawnow;
end
faces = insertObjectAnnotation(InputImage, 'rectangle', BB,
'Face');
figure(6)
imshow(faces), title(['Number of Detected Faces in Input Image
is: ', num2str(n)]);
InputImage = imread(str);
fprintf('Program paused. Press enter to continue.\n');
pause;
else
    figure(6)
    print('No face detected');
end
del = size(size(InputImage));
```

```matlab
if del(:, 2) == 2
   myImg = double(InputImage);
elseif del(:, 2) == 3
    myImg = rgb2gray(InputImage);
    myImg = double(myImg);
end
%% Normalize the Image:
myRange = getrangefromclass(myImg(1));
newMax = myRange(2);
newMin = myRange(1);

InputImage = (myImg - min(myImg(:)))*(newMax - 
newMin)/(max(myImg(:)) - min(myImg(:))) + newMin;


%% continue

figure(7)
subplot(1,2,1)
imshow(InputImage); colormap('gray');title('Input 
image','fontsize',18)
InImage=reshape(double(InputImage)',irow*icol,1);
temp=InImage;
me=mean(temp);
st=std(temp);
temp=(temp-me)*ustd/st+um;
NormImage = temp;
Difference = temp-m;

p = [];
aa=size(u,2);
for i = 1:aa
    pare = dot(NormImage,u(:,i));
    p = [p; pare];
end
ReshapedImage = m + u(:,1:aa)*p;    %m is the mean image, u is 
the eigenvector
ReshapedImage = reshape(ReshapedImage,icol,irow);
ReshapedImage = ReshapedImage';
%show the reconstructed image.
subplot(1,2,2)
imagesc(ReshapedImage); colormap('gray');
title('Reconstructed image','fontsize',18)


fprintf('Program paused. Press enter to continue.\n');
pause;
```

```
InImWeight = [];
for i=1:size(u,2)
    t = u(:,i)';
    WeightOfInputImage = dot(t,Difference');
    InImWeight = [InImWeight; WeightOfInputImage];
end

% Find Euclidean distance
e = [];
for i=1:size(omega,2)
    q = omega(:,i);
    DiffWeight = InImWeight-q;
    mag = norm(DiffWeight, 1);
    e = [e mag];
end
figure(8)
kkk = 1:size(e,2);
plot(kkk,e)
str = 'Image Positions';
xla = xlabel(str);
set(xla, 'Color', 'black')
st = 'Distance between Input imaage and Training images';
yla = ylabel(st);
set(yla, 'Color', 'black')
title('Euclidean distance of input image','fontsize',14)

%% Getting the image with the smallest distance
[emin, ein] =  min(e);
emax = max(e);
%threshold = 0.8 * (emax - ein)
threshold = 3.1737e+04;
if ein < threshold
new = strcat(int2str(ein),'.jpg');
figure(9)
imshow(new);
title('Detected Face','fontsize',14)
else
    fprintf('Face not in database')
end
%% calculating the distance between the detected face and the
closest search
ee = sort(e);
dis = ee(2) - ee(1)
```