FACE RECOGNITION WITH PARTIAL OCCLUSION USING WEIGHING AND IMAGE SEGMENTATION

by

Tapfuma Chanaiwa

Submitted in partial fulfillment of the requirements for the degree
Master of Engineering (Computer Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

October 2020

SUMMARY

FACE RECOGNITION WITH PARTIAL OCCLUSION USING WEIGHING AND IMAGE SEGMENTATION

by

Tapfuma Chanaiwa

Supervisor(s):	Prof. H. C. Myburgh
Department:	Electrical, Electronic and Computer Engineering
University:	University of Pretoria
Degree:	Master of Engineering (Computer Engineering)
Keywords:	Face recognition, partial occlusion, image segmentation, sub-region weighing functions, modified linear discriminant analysis.

This dissertation studied the problem of face recognition when facial images have partial occlusions like sunglasses and scarfs. These partial occlusion lead to loss of discriminatory information when trying to recognise a person's face using traditional face recognition techniques that do not take into account these short comings. This dissertation aimed to fill the gap of knowledge. Several papers in literature put forward the theory that not all regions of the face contribute equally when discriminating between different subjects. They state that some regions of the face are more equal than others, like the eyes and nose. While this may be true in theory there was a need to comprehensively study this problem.

A weighing technique was introduced that that took into account the different features of the face and assigned weights for the different features of the face based on their distance from the five points that were identified as the centre of the weighing technique. Five centres were chosen which were the left eye, the right eye, the centre of the brows, the nose and mouth. These centres perfectly captured where the five dominant regions of the face where roughly located. This weighing technique was fused with an image segmentation process that ultimately led to a hybrid approach to face recognition.

Five features of the face were identified and studied quantitatively on how much they influence face recognition. These five features were the chin (C), eyes (E), forehead (F), mouth (M) and finally the nose (N). For the system to be robust and thorough, combinations of these five features were constructed to make 31 models that were used for both training and testing purposes. This meant that each of the five features had 16 models associated with it. For example, the chin (C) had the following

models associated with it; C, CE, CF, CM, CN, CE, CEM,CEN, CFM, CFN, CMN, CEFM CEFN, CEMN, CFMN and CEFMN. These models were put in five different groupings called Category 1 up to Category 5. A Category 3 model implied that only three out of the five features were utilised for training the algorithm and testing. An example of a Category 3 model was the CFN model. This meant that this model simulated partial occlusion on the mouth and the chin region. The face recognition algorithm was trained on all these different models in order to ascertain the efficiency and effectiveness of this proposed technique. The results were then compared with various methods from literature.

LIST OF ABBREVIATIONS

2DICA	Two-dimensional independent component analysis
2DPCA-ICA	Two-dimensional PCA-ICA
3DCNN	Three-dimensional convolutional neural network
ALBP	Advanced local binary pattern
BDPCA + LDA	Bi-directional PCA plus LDA
BSS	Blind source separation
DCT	Discrete cosine transform
DLBP	Dominant local binary pattern
DT-CWT	Dual-tree complex wavelet transform
FFNN	Feed forward neural network
FHMM	Fuzzy hidden Markov models
FLD	Fisher's linear discriminant
FRS	Face recognition system
Garbor-SVD	Garbor singular value decomposition
HMM	Hidden Markov model
ICA	Independent component analysis
JADE	Joint approximation diagonalisation of eigenmatrices
JAFFE	Japanese female facial expression database
KLT	Karhunen-Loeve transform
KPCA	Kernel PCA
LBP	Local binary pattern
LDA	Linear discriminant analysis
LDP	Local derivative pattern
LGBP	Local Garbor binary patterns
MDRTL	Mean distance regularised triplet loss function
NN	Neural network
OMP	Orthogonal matching pursuit
ORL	Olivetti Research Laboratory database of faces
PCA	Principal component analysis
PDBNN	Probabilistic decision-based neural network
PSVM	Partial support vector machine
PUDBNN	Posterior union model decision-based neural network

PUM	Posterior union model
NMF	Non-negative matrix factorisation
RBFN	Radial basis function network
ReLu	Rectified linear unit
SOBI	Second order blind identification
SPCA-Net	Stacked principal component analysis convolutional neural network
SRC	Sparse representation-based classification
SURF	Speeded up robust features
SVD	Singular value decomposition
SVM	Support vector machine
TBE-CNN	Trunk based ensemble convolutional neural network
UMIST	University of Manchester Institute of Science and Technology database
WPCA	Wavelet principal component analysis
ZDCT	Zoned discrete cosine transform

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1 PROBLEM STATEMENT

1.1.1 Context of the problem

Face recognition is the process of uniquely identifying or verifying a person's identity. This is accomplished by measuring the differences and examining a set of features from a dataset and the image in question. Face recognition has three major steps associated with it namely; image pre-processing, feature extraction and selection, and feature classification. Image pre-processing is the step that standardises and normalises images before feature extraction takes place. Several techniques exist that improves the image quality in the pre-processing phase such as face detection and cropping, histogram equalisation, wavelet transforms, colour normalisation, image denoising and filtering [1–3]. Feature classification is the step that categorises a query image into the right class.

The most vital stage for any face recognition system is the feature extraction stage. This stage extracts and selects the most useful features to be used for discrimination and discards those that are perceived to be of little or no value since some features contribute negatively during the classification stage [4]. Feature extraction also plays a large part in reducing the dimensionality of the input data, in this case the input data is a query image, fed into the extraction algorithm by transforming the input data into a feature set or feature vector [5–7]. Thus, extracting the right features and discarding unnecessary features is of paramount importance.

The performance of a face recognition system drastically degrades under non-ideal conditions such as illumination variation, distance from the observer, face position, age change, and especially partial occlusion. Appearance-based face recognition algorithms are those that use the features on a face such as the eyes, nose, mouth, chin and forehead [8]. If these features are partially occluded with eye glasses or a scarf, this leads to loss of useful discriminatory information that could be used for correct classification [5], [8].

Various appearance-based algorithms exist that use these features when classifying partially occluded images, for example, principal component analysis (PCA), linear discriminant analysis (LDA), local

Gabor binary patterns (LGBP), hidden Markov model (HMM), independent component analysis and speeded up robust features (SURF) [9–14]. These algorithms perform less efficient on partially occluded images as they use face features, thus there is a need to optimise these algorithms such that they can extract only the relevant and available features on a face and in turn discard the occluded regions of a face.

1.1.2 Research gap

The focus of this research was to develop a hybrid model that utilises a different feature extraction algorithms and a dedicated feature classification algorithm. The feature extraction algorithms include PCA and LDA, HMM and SURF. It has been found out that some regions on the face offer little to no discrimination when classifying an image under occlusion [5].

There is a need to implement an algorithm that can assess and quantify which regions of the face, and combinations of regions as well, which offer the best discrimination. Some regions of the face are better at discriminating one individual from another, thus this research aims to weigh these different regions according to what extent they influence discrimination and to what extent they do as well.

1.2 RESEARCH OBJECTIVE AND QUESTIONS

The image will be segmented into modular sub-regions and then weighing each of these regions to find the right combination that leads to optimal performance. Some of the sub-regions would then be discarded to increase performance of the hybrid model. Some research questions that arose are:

- Is it possible to develop a hybrid face recognition model using a combination of different feature extraction algorithms and a single classification algorithm that can outperform the individual algorithms?
- Can segmenting an image into smaller modular sub-regions and then applying a feature extraction algorithm to each of those sub regions lead to higher recognition rates as compared to applying the feature extraction algorithm onto the entire image?
- Once an image has been segmented into smaller regions, is it possible to discard some sub regions which do not offer much discrimination and simultaneously lead to an efficient recognition system?

1.3 HYPOTHESIS AND APPROACH

1.3.1 Hypothesis

1.3.1.1 I. Hybrid hypothesis

This study aims to assess the hypothesis that using a hybrid model that consists of a combination of two techniques in the feature extraction and processing phase, leads to better performance in terms of recognition rates.

1.3.1.2 II. Apply feature extraction to sub-regions hypothesis

It is hypothesised that applying a feature extraction technique on sub-regions of an image leads to a better performing recognition system as compared to using the entire image. Thus using a weighted approach to these sub-regions leads to an increase in efficiency of the face recognition system.

1.3.1.3 III. Image segmentation hypothesis

This study aims to assess the hypothesis that segmenting an image into smaller sub-regions and discarding the sub-regions that offer the least discriminatory information can lead to a more efficient and better recognition system.

1.3.2 Procedure

1.3.2.1 I. Hybrid approach

This dissertation aims to tackle the problem of improving improving facial recognition rates when the subject is partially occluded. This can be accomplished by combining different feature extraction techniques to improve recognition rate.

It has been shown that recognition rates increase when using a hybrid method as compared to just using a single algorithm for both feature extraction and feature classification [5–7], [12], [15], [16–18]. It should be possible to use a hybrid method that utilises LDA as a feature extraction algorithm and a feed forward neural network (FFNN) as a feature classifier.

It has been shown that using FFNN, as a feature classification algorithm, is feasible and leads to a recognition rate of above 94% [18]. It was also shown that using LDA as a feature extraction technique leads to a more efficient facial recognition system [19]. The performance of this hybrid model can be compared to both traditional LDA and FFNN when they are used individually as both feature extraction algorithms and feature classification algorithms. In order to assess the performance several databases that contain images under partial occlusion are used as training and testing purposes. 30% of the images will be used for testing purposes and 70% of the images will be used for training purposes. K-cross fold validation will be used to assess the accuracy of the proposed hybrid method. K-cross

fold validation entails cycling through an entire dataset of faces in order to make sure all the faces have been used for both training and testing purposes. Data used in this experiment will be collected from face recognition databases such as AR database, ORL database, Yale Faces database and UMIST database. These databases are readily available online.

1.3.2.2 II. Image segmentation approach

Some literature divides an image into four equal modular sub-regions without taking into account where the key features are located [20]. Research will be done to find out the optimal number of sub-regions an image can be segmented into whilst retaining a high recognition accuracy. Different combinations of the sub-regions will also be tested in order to find the combination of regions that provide optimal efficiency with the hybrid model. This can be tested by segmenting faces in different face databases and assessing the accuracy of the model by utilising K-cross fold validation.

In order to assess the performance several databases that contain images under partial occlusion are used as training and testing purposes. 30% of the images will be used for testing purposes and 70% of the images will be used for training purposes. Both raining and testing dataset would contain images with and without occlusion. K-cross fold validation will be used to assess the accuracy of the proposed hybrid method. K-cross fold validation entails cycling through an entire dataset of faces in order to make sure all the faces have been used for both training and testing purposes. Data used in this experiment will be collected from face standard recognition databases namely the AR database ORL database, Yale Faces database and UMIST database. These databases are readily available online.

This approach has a possible failing point as well, which is the image may be segmented into a large number of sub-regions which would inversely lead to a drastic decline in performance as it would take a large amount of time to process all these sub-regions. It can also fail when these sub-regions are segmented at points where they cut the features which offer the most discrimination such as the eyes, nose and mouth thus unintentionally leading to a greater loss of information.

For any research to be successful a strong grasp and understanding of the theory behind the literature that underpins it is of paramount importance. Similarly, for this research into feature extraction techniques to be successful a strong understanding of feature extraction algorithms will be obtained by reading extensively journal papers and organising them into credible sources. Research will have to be performed to in order to mathematically model these algorithms and turn that mathematical model into pseudo-code which would be used to code the algorithms. The algorithms will then be run on different databases in order to ascertain their accuracy.

1.4 RESEARCH GOALS

This research aims to achieve the following:

- To develop a hybrid algorithm that fuses different feature extraction techniques in order to increase face recognition metrics.
- To propose the right combination of features that leads to a higher recognition rate.
- To develop the right weighing for each sub-region that can lead to a higher recognition rate.

1.5 RESEARCH CONTRIBUTION

This dissertation makes the following contributions:

- A weighing technique is presented that makes use of five centres on a human face. These five centres act as the point of reference when weighing the features used for recognition. These centres were located at the centre of the left eye, the centre of the right eye, the centre of the brow, the centre of the nose and the centre of the mouth. A weighing function was then applied with this in mind.
- Presents an image segmenting method that divides the face into five sub-regions used as features for face recognition. These features are made into thirty-one possible combinations used for face recognition. From these combination it was found out that the forehead (F) feature and its 15 other models offered the most discrimination.
- Makes a comparative study on which parts of the face offer the most discrimination during face recognition.
- A hybrid linear discriminant analysis algorithm that utilises segmentation and weighing techniques.

1.6 RESEARCH OUTPUTS

The following article has been submitted for publication:

T. Chanaiwa, and H.C. Myburgh, "Modified Linear Discriminant Analysis using Image Segmentation and Weighing Techniques," *Elsevier Journal on Image and Vision Computing*. Submitted.

1.7 OVERVIEW OF STUDY

This study is organised as follows:

Chapter 2 offers a comprehensive literature survey of the past and present algorithms used for a facial recognition system (FSR). It underlines the theoretical background of how these methods function.

Chapter 3 is composed of the algorithms that were implemented as a means to compare them against the proposed technique under the same conditions. These algorithms include Discrete Cosine Transform (DCT), Independent Component Analysis (ICA), Non-negative Matrix Factorisation (NMF) and Sparse Representation-based Classification (SRC).

Chapter 4 focuses on the proposed research and how it was implemented. In this dissertation a face segmentation and weighing algorithm was proposed that divides the face into regions and examined which parts offered the most discrimination. The parts that offered the most discrimination were given higher weights so as to increase discrimination. This resulted in a modified linear discriminant algorithm that utilised segmentation and weighing.

Chapter 5 consists of a presentation of the results obtained from the proposed technique. It also compares the results taken directly from literature and the results obtained from experimentation. These results are discussed in depth to so as to provide their contextual significance.

Chapter 6 concludes and summarises this dissertation. It offers a brief overview of what was done, the key results obtained, the significance of the results obtained, the contribution of the author and suggestions are presented for future works.

CHAPTER 2 LITERATURE STUDY

2.1 CHAPTER OVERVIEW

In Section 2.2 an overview of how a facial recognition system functions is presented. The three different stages of the system are introduced concisely. In Section 2.3 looks at the history of face recognition, the early remnants of any semblance of a face recognition system, its inception, the different stages it went through from man-machine to a fully automated approach. Section 2.4 discusses image pre-processing in depth, it presents two different techniques that are used in the pre-processing techniques, mainly greyscale conversion and histogram equilisation. Section 2.5 presents different techniques that are used for feature extraction. These techniques are good at selecting useful features that can be used for recognition purposes. These techniques include Gabor wavelet transform, speeded-up robust features, hidden Markov models and Hough transforms. In Section 2.6 different techniques that directly aid in feature classification are presented. These techniques include neural networks, support vector machines and nearest neighbour techniques.

2.2 OVERVIEW OF THE FACE RECOGNITION SYSTEM

The problem of face recognition has been extensively studied throughout the past three decades. Human beings recognise faces quite easily on a daily basis. This process has become automatic for human beings, as it has been debated that humans start recognising faces as a tender age of nine months. Humans automatically can classify a face by the main features of the face namely; the eye region and nose region. Machines, on the other hand, have a difficult time recognising faces as compared to humans. A face recognition system that can automatically detect faces has mainly three stages. Figure 2.1 shows what a typical face recognition scheme is like.

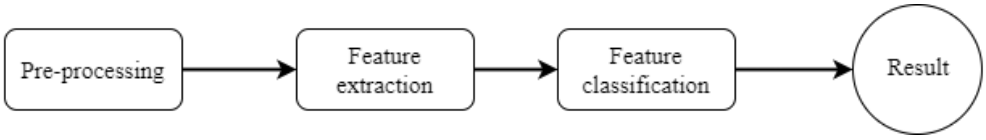


Figure 2.1. A typical face recognition scheme showing the four key stages.

2.2.1 Image pre-processing

This entails normalising an image by removing any unwanted distortions that may hinder the recognition process, enhancing an image's intensity or contrast in order to improve accuracy. Several techniques exist that improves the image quality in the pre-processing phase, such as face detection and cropping, histogram equalisation, wavelet transforms, colour normalisation, image de-noising and filtering [18], [21].

2.2.2 Feature extraction

The most vital stage for any face recognition system is the feature extraction stage. This stage extracts and selects the most useful features to be used for discrimination and discards those that are perceived to be of little or no value since some features contribute negatively during the classification stage [22], [23]. Feature extraction also plays a large part in reducing the dimensionality of the input data, in this case the input data is a query image, fed into the extraction algorithm by transforming the input data into a feature set or feature vector [24], [25]. Thus, extracting the right features and discarding unnecessary features is of paramount importance. There are various methods that have been used to do feature extraction on images containing faces under partial occlusion. These methods include Discrete Cosine Transform (DCT), Garbor Wavelet Transform (GWT), Hough Transform (HT), Speeded Up Robust Features (SURF), Scale Invariant Feature Transform (SIFT), Local Binary Pattern (LBP) and Elastic Bunch Graph Matching (EBGM), Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) [26], [27].

PCA, also referred to as the eigenface method for face recognition, is a statistical method that reduces the dimensionality of data by calculating a set of orthogonal vectors from the training samples [28]. It linearly transforms the original image space into an orthogonal eigenspace with reduced dimensionality [29]. A query image can be approximated using a weighed combination of the eigenfaces [28].

LDA, on the other hand, linearly transforms the input data by seeking to maximise the between class scatter and minimising the within class scatter [30]. This is done so as to make the images in one class cluster in the smallest region possible and maximising the distance between the means of different classes when projected into a lower subspace [30].

The EBGM algorithm represents input images of human faces as graphs, nodes and edges labelled with distance vectors. These nodes consist of Gabor wavelet coefficients which encode the grey value distribution of an image [31]. The edges encode the geometry of the given face. A graph is chosen such that it maximises the graph similarity function.

LBP feature extraction was implemented and the corresponding output feature vectors were compared. LBP is a texture based face recognition algorithm, wherein it is capable of efficiently describing both the shape and texture of images in digital format [26].

2.2.3 Feature classification

Face recognition has come a long way since the early 1980's when it was semi-automatic. A human had to mark the key points on an image so that a machine could calculate the distances between those points. Once the distances were computed the machine would simply select a corresponding picture which closely matched the query image. Selecting the closest image was an earliest form of classification. Formally speaking, classification is a technique in statistics and machine learning where a computer program is fed input data, and the program learns from the data and uses the data to classify new and unlearned observations. The input data can vary from being bi-class or multi-class [32].

Bi-class data is data that only consist of two distinct classes of observations (identifying whether it is raining or if it is sunny). Multi-class data is data which has more than two distinct classes of data. When applied to face recognition, these classes can be different people. Classification plays a major role in recognition problems such as handwriting, speech, ear, palm print, fingerprint and face recognition [32], [33]. Feature classification can also be used for face verification. This is a scenario of trying to match a query image to a single training image rather than an entire training database that may contain more than 1000 unique images. There are several feature classification techniques such as, support vector machines (SVM), naive Bayes classifier, logistic regression, neural network, decision trees, random forests, chi square, and nearest neighbour classifier [22], [23], [34], [35]. SVMs are a statistical model that projects input data into a higher dimensional space and computes an optimal hyperplane that is capable of discriminating multi-class data [36].

2.3 BRIEF HISTORY ON FACE RECOGNITION

During the mid 1960's Wilson Woodrow Bledsoe developed a machine that could recognise faces with efficiency greater than what humans are capable of. This led to the development of what was known as a "man-machine" face recognition system where a human [37]. In order to recognise faces, a picture would be taken of a human face align it with a pre-defined template and add coordinates to the face. The coordinates were manually added in order to mark the position of the centre of the pupils, the inside and outside corners of the eyes, the tip of the nose, the middle of the lips, the edges of the lips, and the widows peak [37].

The coordinates that were marked were entered into a machine. The machine would calculate twenty distances such as the width of the mouth and eyes, the length of the nose and the distance between the

left eye and the centre of the mouth. These were known to be the discriminating features between two different individuals. The machine was able to calculate these distances at a rate of 10 pictures every 15 minutes. When faced with a query image the coordinates of the key points were entered into the machine and the machine would output the closest match to the entered coordinates, in essence that is how it was able to recognise faces. The machine lacked the robustness to different conditions, in Bledsoe's own words:

"This recognition problem is made difficult by the great variability in head rotation, tilt, lighting intensity, light angle, facial expression, ageing, etc. Some other attempts at face recognition by machine have allowed for little or no variability in these quantities. Yet the method of correlation (or pattern matching) of unprocessed optical data, which is often used by some researchers, is certain to fail in cases where the variability is great. In particular, the correlation is very low between two pictures of the same person with two different head rotations."

Instead of using 40 or more coordinates to input into the machine experiments were done in order to ascertain which features discriminates the most between different individuals. The experiments used a pool of 10 jurors (5 males, 5 females) who selected the features they thought provided the most variability among 256 test pictures. Statistical modelling was then done to reduce the bias of the jurors in selecting the features. In the end, a total of 21 specific subjective markers which included but not limited to hair colour, the thickness of lips, the shape of the head, the width of the eyes to input into the machine for recognition [38], [39]. There were some improvements in the accuracy and speed of face recognition however the process was still semi-automatic [38]. Ultimately, this method was just as slow and semi-automatic as they relied on a human operator to first mark the features and then input them into a machine.

In the late 1980's the goal of recognition using purely mathematics was accomplished [40]. A machine would just have to be presented with a cropped picture of a human face. Since a picture of say 100 by 100 pixels would contain 10 000 pixels that needed to be evaluated, using linear algebra, specifically principal component analysis (PCA), in a quest to reduce the dimensionality from a subspace of 10 000 into a subspace of less that 50. This reduction in dimensionality meant the time taken for recognition vastly improved. This reduction in dimensionality became known as the Eigenface approach. The Eigenface approach used eignevectors derived from a covariance matrix of a probability distribution. They were selected in such a way that the eignevectors with the most variance produced the greatest discrimination among the test subjects [40]. The chosen eigenvectors also known as eignefaces formed the basis of the features used in both the testing and training stage. A linear combination of these some N eigenvectors when reconstructed would yield the closest match in the training set [40].

The method of using purely mathematics to solve the recognition problem was expanded into using the eigenface approach to reduce the dimensionality of an image as well. In so doing, it proposed a new method that was able to automatically detect the human face, crop it and then feed it as input into the recognition system [28]. The test image was projected onto a low-dimension feature space that has eigenvectors with the most variations [28]. This was the first time that the process of face recognition achieved full automation and it was near-real time.

In order to increase accuracy, the mathematical approach and the eigenface approach was expanded by adding 2 additional steps that increased the recognition accuracy [41]. Linear discriminant analysis (LDA) was used to compute the covariance matrix that was used in projecting the face from a higher dimension to a lower sub space of features. During the projection step they used Fisher's liner discriminant (FLD) in order to increase the between class scatter and minimise the within class scatter of any set of images [41]. In order to have more separation when the classes were projected into a lower dimension all the images pertaining to any distinct subject were purposely marked as belonging to that subject. This enable the algorithm to then apply FLD [41].

Hidden Markov models were extensively studied for the recognition process [42–44]. An approach that automatically extracts the key identifying features of an image and uses them to recognise a test subject was put forward [42]. In the model proposed an input image was divided into N horizontal overlapping strips, on each strip a sliding window would generate a spatial sequence that will in turn be used in the training of the HMM [42]. The HMM would partition the spatial sequence into a feature state [42–44]. This model has primarily 2 matrices, the transition state matrix denoted by **A** and the output probability matrix denoted by **B**. Once the HMM was trained both matrix **A** and **B** would represent different features of the original image. The occurrences of different transitions on the entire face and thickness of the feature are populated in matrix **A**. Matrix **B** represents the simple probability of remaining in the same state.The combination of these 2 matrices provides strong discrimination between different test subjects.

Local binary patterns histograms were also used for recognition [45]. This is a texture descriptor method that classifies texture and finds the occurrence of LBP codes in an image, these codes are then collected into a histogram. These histograms are obtained from different images that contain information about facial micro patterns which include the location of eyes, the edges of a face, the location of the nose and mouth and the flat areas of the face that enable full representation of the facial features [45]. Classification is achieved by computing similarities between the different histograms.

A convolutional neural network (CNN) consists of multiple layers; the input layer, the pooling layer, the convolutional layer, the non-linearity layer, the fully connected layer and the output layer [46], [47].

The input layer is simply the image that needs to be recognised. The hidden layer consists of multiple layers that each have neurons that are activated when given an input. Each neuron receives input and has an individual weight associated with it [46]. The convolutional layer and non-linearity layer does computations and then outputs the response to the output layer whether the input image has been recognised or not.

2.4 IMAGE PRE-PROCESSING STAGE

The accuracy and reliability of any face recognition system can be affected by a number of factors which include; inconsistent and drastic light intensity, non-neutral facial expression, pose variations, degree of tilt of the head and occlusions (glasses, hat, mouth cover etc). The biometric system was designed with the intention of it functioning rapidly when detecting and recognising a person. To meet this speed objective there is a need to speed up that process by removing any systematic differences that affect different images. Image pre-processing then comes into play so that it improves the data embedded in an image and enhance the features needed by the detection, feature extraction and recognition stages.

2.4.1 Terminology

2.4.1.1 Defining a Colour Image

All colour images are some variation of the red, green, blue (RGB) model. This model defines all the colours as combinations of three primary colours namely red, green and blue. Each pixel in any image has three channels which combine to constitute the single colour of that specific pixel. All pixels in an image can be represented as an RGB triplet [R,G,B] of which all the colours are integers that range from 0 to 255 of that specific colour, 0 being absolute black and 255 being absolute white.

Table 2.1. RGB Triplet Representation

Colour	RGB
White	[255, 255, 255]
Black	[0, 0, 0]
Red	[255, 0, 0]
Green	[0, 55, 0]
Blue	[0, 0, 255]
Yellow	[255, 255, 0]
Magenta	[255, 0, 255]
Cyan	[0, 255, 255]

2.4.1.2 Defining a Greyscale image

A greyscale image is an image that has only one intensity value per pixel as compared to three for a colour image. This intensity value of the pixel ranges from 0 to 255, 255 being white and 0 being absolute black and the rang 1 to 254 is just 254 shades of grey. A black and white image only contains 2 colours black and white, thus it can be represented by just 1 bit; either 0 or 1.

2.4.2 Greyscale Conversion

Using greyscale images as compared to using RGB images has the advantage of speeding up processing time 3-fold. This is because RGB images are described by 3 intensity values for any pixel. On the other hand, a greyscale image is only described by 1 intensity value per pixel. This can be seen as compressing the data contained in a RGB image. This information loss is insignificant to the detection and recognition phase as a recognition method can easily recognise a face in greyscale format. Although there exist various methods to convert from RGB to greyscale for example, simply averaging all 3 pixels and making them 1 value:

$$\text{Gray} = (\text{Red} + \text{Green} + \text{Blue}) / 3. \tag{2.1}$$

However, this method doesn't take into account the intensity the primary colours contribute to a greyscale image. So to convert to greyscale a method which takes into account the luminance of the colour image was used. This luminances simply describes how our human eyes sensitive to the entire wavelength of visible light modelled according to the ITU-R Recommendation BT.709 standard [48] . This method uses the following formula to convert an image from RGB spacial representation into greyscale representation yielded Figure 2.2 [49]:

$$\text{Grey} = (\text{Red} \cdot 0.299 + \text{Green} \cdot 0.587 + \text{Blue} \cdot 0.114) / 3. \tag{2.2}$$

(a) RGB image (b) Greyscale image

Figure 2.2. Converting an RGB image to a greyscale image..

2.4.3 Histogram Equalisation

The biometric system is required to function in conditions with varying light intensity. Thus taking images when there are major variations in light intensity means that some images would have a lower contrast between their pixels and other images would have a higher contrast between their pixels. This in turn would affect the efficiency of the biometric system. Histogram equilisation was utilised to counteract the effect of differing contrast from one image to another. The distribution of the greyscale pixels in images in the databases was non-uniform. Therefore, histogram equilisation was used to normallise the image by using a non-linear monotonic mapping to change the values of the pixel intensity so that the distribution of the entire pixels becomes uniform [50]. This had the consequence of making the pixel with the lowest intensity pitch black (that means it was assigned 0 on the RGB model) and conversely the pixel with the highest intensity was made white (that means it was assigned 255 on the RGB model). Hence all pixel areas with low contrast were made into high contrast [50].

Say there is a 50 x 50 greyscale image defined by the vector x_i, this means any pixel in that image is defined as $x_{i,j}$ where j ranges from 1, .., 50. All the pixels in the image are discrete integers where each pixel is an element of $x_{i,j} \in [0,255]$. Therefore, it follows that when the probability of pixel $x_{i,j}$ has a probability of c, the Probability Distribution Function (PDF) of the entire image is given by:

$$p(c) = \frac{1}{N} \sum_{j=1}^{N} P\left(\mathbf{x}_{i,j} = c\right). \tag{2.3}$$

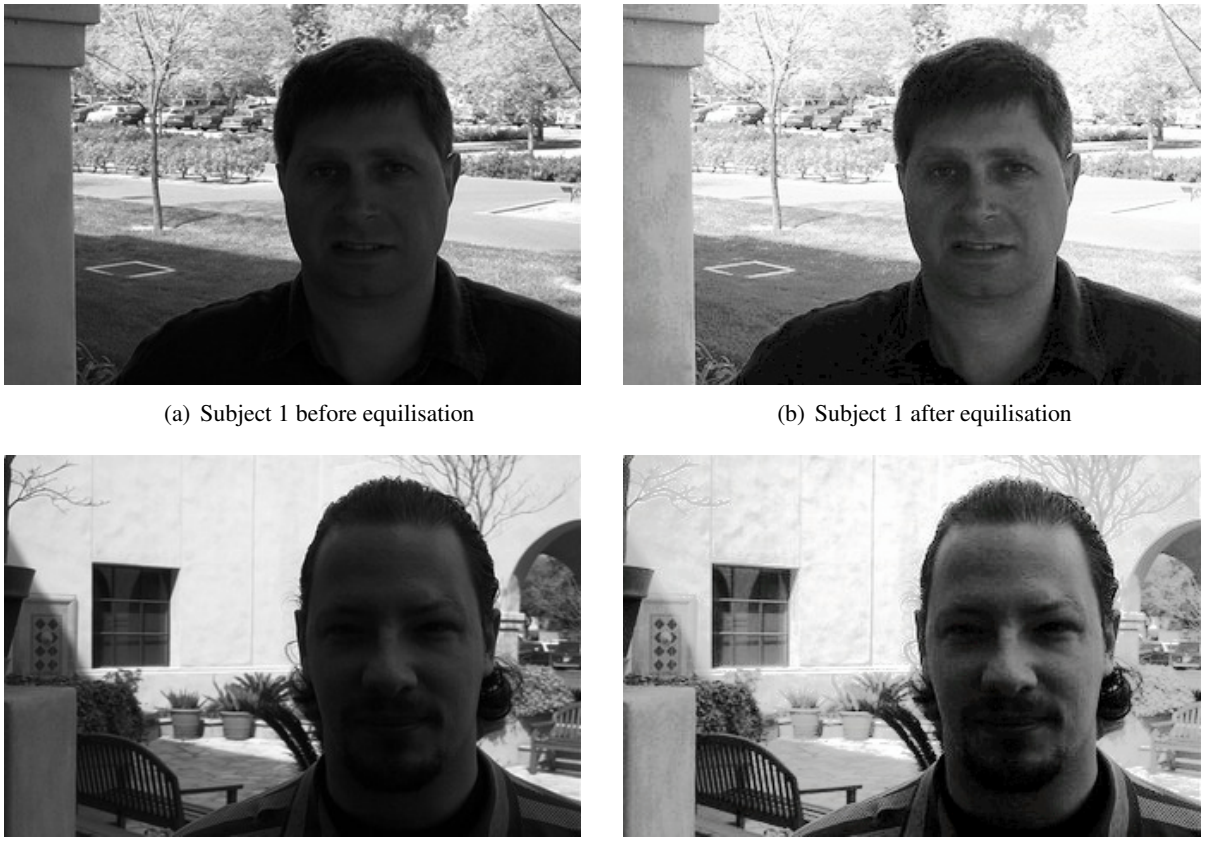
For a specific grey level c, the image's histogram was nomalised to be an element of $\in [0,1]$. In turn, the cumulative distribution function (CDF) was calculated from the image's PDF using:

$$\mathrm{CDF}(c) = \sum_{d=0}^{c} p(d). \tag{2.4}$$

Figure 2.3 shows the effect how histogram equilisation improved the contrast of two images. As can be noted in Figure 2.4, the output histograms are almost uniform in nature. However, the resulting CDF is linear. This linearity means that both images have been successfully equilised. The aim is to equally represent all grey levels in any image indiscriminately. To achieve this a liner transformation was applied according to (2.5):

$$h(i) = \mathrm{round}\Big(\tfrac{\mathrm{CDF}(c) - \mathrm{CDF}_{\mathrm{min}}}{\mathrm{N} - \mathrm{CDF}_{\mathrm{min}}} \times (L-1)\Big), \tag{2.5}$$

where, $\mathrm{CDF}_{\mathrm{min}}$ was the minimum grey level intensity in any given image. $\mathrm{CDF}_{\mathrm{max}}$ was the maximum grey level intensity in any given image and L was the number of grey level taken on by the pixels, 256. Equation (2.5) ensured that all images were normalised and their grey level started from 0 and ended at 255. All images for detection and recognition purposes underwent histogram equilisation.



(a) Subject 1 before equilisation

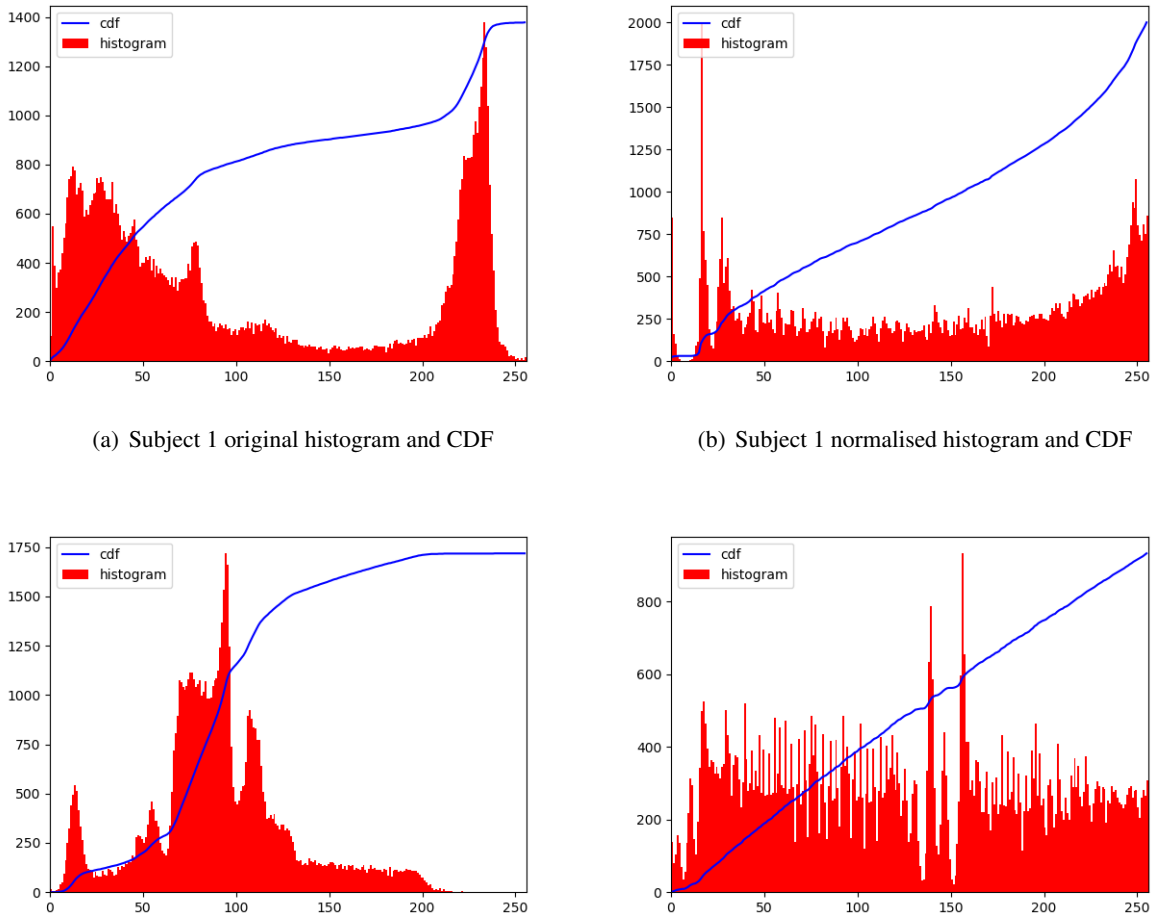
(b) Subject 1 after equilisation

(c) Subject 2 before equilisation

(d) Subject 2 after equilisation

Figure 2.3. Histogram equilisation effect on images.

Figure 2.3 shows how low contrast images were converted to high contrast images through histogram equilisation. It can be seen that the original images lacked contrast.



(a) Subject 1 original histogram and CDF

(b) Subject 1 normalised histogram and CDF

(c) Subject 2 original histogram and CDF

(d) Subject 2 normalised histogram and CDF

Figure 2.4. CDF and Histograms before and after equilisation.

Figure 2.4 (a) shows that the majority of the pixels from subject 1 were in the 0-90 and 230-250 region which meant the image lacked contrast. Likewise Figure 2.4 (c) the majority of subject 2's pixels were in the 50-120 range which also meant the image had less contrast. Figure 2.4 (b) and (d) show that the pixels were distributed almost evenly throughout the entire 0 to 255 greyscale range. Figure 2.3 shows the effect how histogram equilisation improved the contrast of two images. As can be noted in Figure 2.4, the output histograms are not uniform in nature, however the resulting CDF is relatively linear. This linearity means that both images have been successfully equilised. The aim is to equally represent all grey levels in any image indiscriminately.

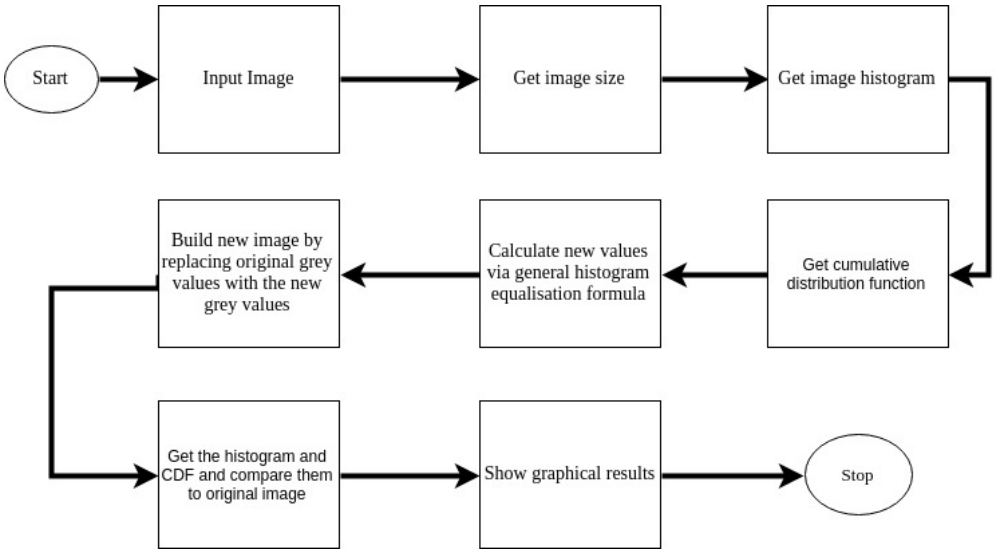


Figure 2.5. Flow diagram for histogram equilisation algorithm

Figure 2.5 shows the flow for the histogram equilisation. The algorithm ran was computationally inexpensive thus it ran in real time. It did not affect the speed of detection and recognition.

2.5 FEATURE EXTRACTION STAGE

2.5.1 Linear Discriminant Analysis

LDA is also known as Fisherfaces as well as "traditional" LDA. The recognition model utilises the Fisherfaces algorithm to extract features, train the model and finally classify the query image. Fisherfaces algorithm is a pattern classification approach which minimises the effects of varying light intensity, poses and different facial expressions [51]. LDA in face recognition is usually done in 10 steps. Fisherfaces ulitises LDA which main aim is to reduce the dimensionality of the training class to a lower dimension while at the same time conserving all the discriminatory information that best separates different classes from each other [51]. LDA selects some features or discriminants that do the best job of separating the facial images into different classes in the feature space. This feature space is the low dimensional space.

The aim is to minimise the variations within a specific class whilst maximising the variations between classes. This means that when images belonging to a single class are projected to the discriminant space they cluster together with minimum separation, while the separation between classes is maximised.

2.5.1.1 Variants of LDA

A. Direct Linear Discriminant Analysis (DLDA)

DLDA takes the linear space of all class means in any given testing and training set, as the final solution to the traditional LDA. In traditional LDA, the covariance matrix is constructed from the maximisation of Fisher's criterion, in the case of DLDA this covariance matrix does not play a role in the computation of the solution [52]. The critical idea is that for any given LDA algorithm, when calculating the between class scatter S_b of the classes, the directions on which s_b form a null space are futile in finding the solution. Thus discarding this null space of S_b reduces the dimensionality of the initial higher dimension by a full rank, from $n \cdot n$ to $C - 1$ Where $n \cdot n$ are the total number of pixels in all the pictures in the training set and C is the total number of classes in the training set [52]. The simultaneous diagonalization step is then performed where by the denominator is minimised while the numerator is whitened [52], [53]. As a result, the null space of the within class scatter S_w is untouched. Similarly the rest of the steps of traditional LDA were performed.

B. Direct Weighed Linear Discriminant Analysis (DW-LDA)

A method that could combine the benefits of DLDA and WLDA was proposed in order to increase the accuracy of the algorithms [54]. In order to harness the the power of both DLDA and WLDA certain assumptions on both algorithms were made in order to make computation much faster. The first assumption they made was that all the means of the different classes were **distinct** thus no two classes had the exact same mean. The second assumption was that the inverse of the within class scatter, S_w^{-1} existed and was real [54]. Similarly the rest of the steps of traditional LDA were performed and consequently achieved an accuracy of 95% [55].

C. Modular Linear Discriminant Analysis (MLDA)

Earlier research had shown that expressions, occlusions or any decoration on the face plays a crucial role in the accuracy of any system [56], [57]. Thus [54] proposed a framework whereby the image containing a face is divided into modular sub-regions. The framework that was developed utilised a system of observers arranged in a parallel formation as in Fig 2.6.

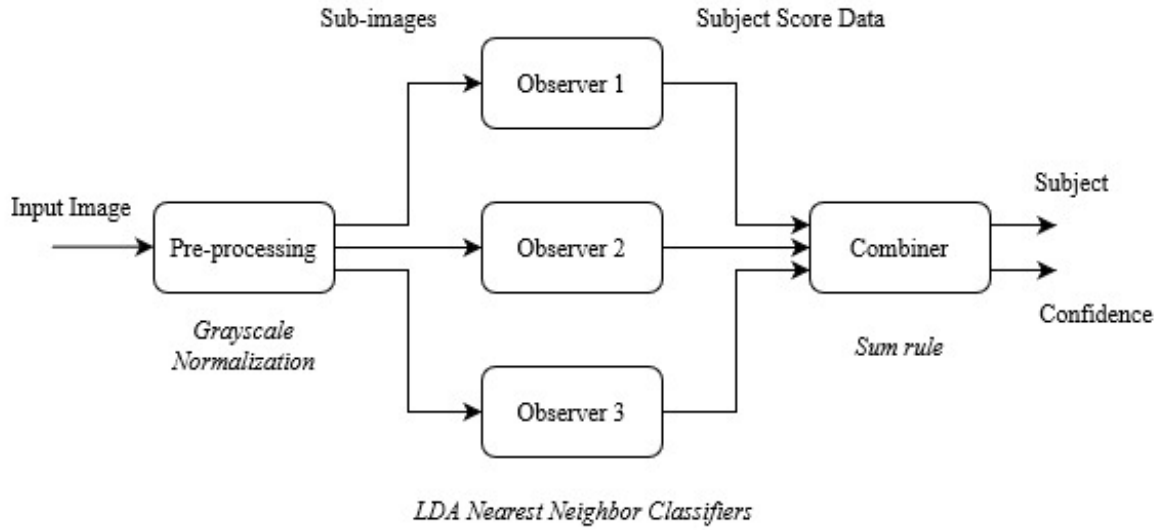


Figure 2.6. Framework of observers [54].

Each observer is used to supervise and perform the training phase of a single sub-region of a face. The combination of all these observers is known as a framework. The output from the entire framework is simply summed to provide the final output classification [54].

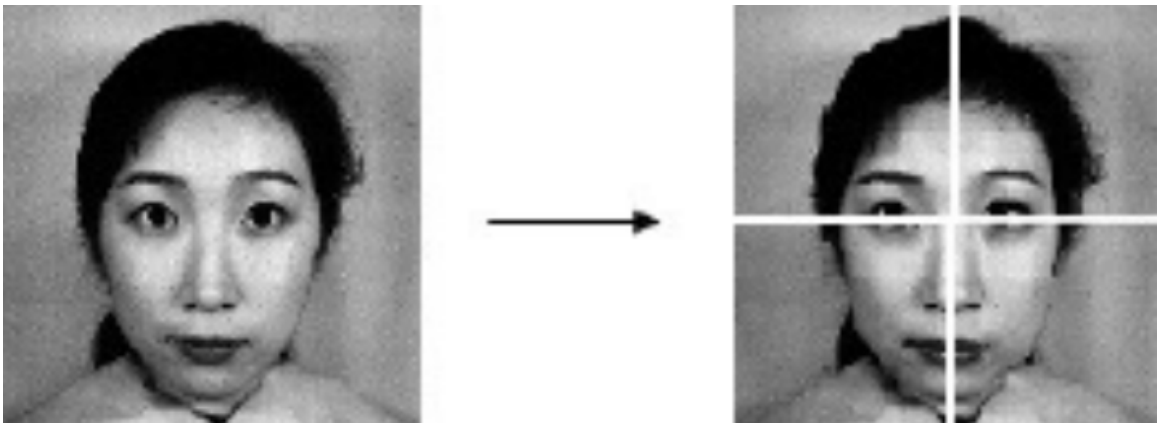


Figure 2.7. Modular face image regions used by the four observers [58]

Figure 2.7 shows an example of how an image may be divided into 4 different sub-regions that would be fed into the MLDA framework. This framework provides robustness to occlusions and is invariant to facial expressions [59]. This is because some of the observers would be trained on sub-regions of the face without any occlusions and decorations.

2.5.2 Hidden Markov Models

A HMM is a statistical model which is utilised in the characterisation of statistical properties if a signal [60]. HMMs consists of two distinct processes, the first one being an underlying and unobservable Markov chain and the second one being probability density functions (PDFs). The Markov chain consists of a restricted number of states, a transition state matrix, an initial probability state matrix [42], [43], [60]. All the states have an associated PDF. Before HMM were used in face recognition they were predominantly used in speech recognition because of their one-dimensional feature extraction nature. In order for them to be adapted to recognition of faces from 2 dimensional images, the images can be modelled as a one-dimensional continuous strip [60]. Since the features of a face, taking the top-down approach, start with the forehead, then eyes, nose, mouth, and finally the chin that meant all these features could be assigned a state in a one-dimensional HMM.

2.5.2.1 Variants of HMM

A. HMM using two-dimensional Discrete Cosine Transform (2D-DCT)

This method differs from the traditional HMM feature extraction in the sense that it does not take direct pixel values for its observation vector since pixel values do not perfectly represent the facial features, are susceptible to change in illumination and occlusions and are very sensitive to any noise in the image [60], [61]. Thus to counter this problem, the sliding window that captures the blocks, is slid throughout the photo in a zig-zag fashion. At each and every sub window, the 2D-DCT is calculated and only the first few DCT coefficients are stored and the rest are discarded. These coefficients made up the observation vector. This method yielded 99.5% recognition accuracy for when using 9 images for training purposes [61].

B. Two-Dimensional Hidden Markov Models (2DHMM)

Images are two-dimensional in nature, thus extracting features using a one-dimensional HMM leads to loss of information that could be used for classification purposes [62]. The paper employed a variety of pre-processing techniques in order to normalise the input face image: cropping face region, cropping face area, scaling the image which results in the distance between the inner corners of the eyes being 120 pixels [62]. For the feature extracted Wavelet Transform was used because it can approximate high-scale components and low-frequency components of a signal. Low pass and high pass filtering was done in order to segment the face into distinct sub images [62]. In the column direction, the sub images were decomposed into the low-pass (L) and high pass (H) frequency bands both these bands were subsequently decomposed along the vertical, diagonal and horizontal axis to become LL, LH, HL and HH [62]. This enable a 2D Markov chains to be formed.

C. Hybrid of a 2DHMM with an artificial neural network

A back propagation neural network (BPNN) can be used to reduce the dimensionality of an image by extracting facial feature components that were key in determining the identity of a person. These components were modelled as coefficients of a BPNN [63]. The 2DHMM was composed of 5 states representing the forehead, eyes, nose, mouth and chin. The coefficients were used to train the 2DHMM. This hybrid model had a recognition rate of 100% [63]. The database used for training and testing purposes was the Olivetti Research Laboratory (ORL) database. The database consisted of frontal face images having variable illumination, poses, gender and occlusions.

2.5.3 Discrete Cosine Transform

Discrete Cosine Transform (DCT) is a method that converts any given signal into its basic elementary components and is mostly unitised when there is a need for image compression [64]. DCT achieves image compression by using 2 important steps. It first quantizes the images DCT coefficients and secondly does entropy coding on the extracted DCT coefficients. Quantization reduces the number of possible values of any given quantity consequently reduces the amount of bits required to represent it. In facial recognition this is dimensionality reduction. DCT can be used in both a holistic and local appearance-based methods. When DCT is used in local appearance based method the image is segmented into blocks of equal area and it is applied to these areas separately, the local features which are extracted are then combined on the feature and decision level. The DCT coefficients obtained from each individual block are then concatenated to make up a feature vector containing the discrimination information for classification [64]. Since the image is segmented into blocks of equal area, decision fusion is done on each block and these results are combined to make the final classification. After DCT is applied to the blocks, it pushes lower frequencies towards the upper left corner of each block and information is extracted by using a zig-zag scan on the blocks.

2.5.4 Garbor Wavelet Transform

Garbor wavelet transform (GWT) are transforms also used in facial feature extraction. GWT analyses images with optimal frequency and spatial resolution. These wavelets are modelled after the response of cells in the mammalian visual cortex and they extract local features for face recognition [65]. The wavelets are defined by the following Gaussian function [66]:

$$\psi_j(\vec{x}) = \frac{k_j^2}{\sigma^2} \exp\left(-\frac{k_j^2 x^2}{2\sigma^2}\right)\left[\exp(j\vec{k}_j\vec{x}) - \exp\left(-\frac{\sigma^2}{2}\right)\right]. \tag{2.6}$$

In order to make the wavelets robust to illumination changes the $\exp(-\frac{\sigma^2}{2})$ term is subtracted. Images can be represented by using the GWT which in turn allows them to be described in terms of spatial

structures and spatial frequencies. When the image is convolved using 40 Gabor filters, with spatial frequency $v = 0, 1, 2, 3, 4$, and different orientations the amplitude and frequency is captured. This is used to generate the Gabor filters. Figure 2.8 shows the family of the 40 Gabor wavelets. For feature extraction, the features are extracted at points that contain high information content for example; the eyes, nose, lips, dimples. The features are extracted when a window, W_0, of a certain area and selecting the maximum value pixel according to (2.7) and (2.8) [66]:

$$R_j(x_0, y_0) = \max_{(x,y) \in W_0} (R_j(x, y)), \tag{2.7}$$

$$R_j(x_0, y_0) > \frac{1}{N_1 N_2} \sum_{x=1}^{N_1} \sum_{y=1}^{N_2} R_j(x, y), j = 1, \ldots, 40, \tag{2.8}$$

where, R_j is the response of any given image to the j^{th} Gabor filter, N_1 and N_2 are the length and the width of the given image, the centre of W_0 is at (x_0, y_0).

An accuracy of 95.25% was achieved on the Purdue database which had faces with partial occlusion [66].

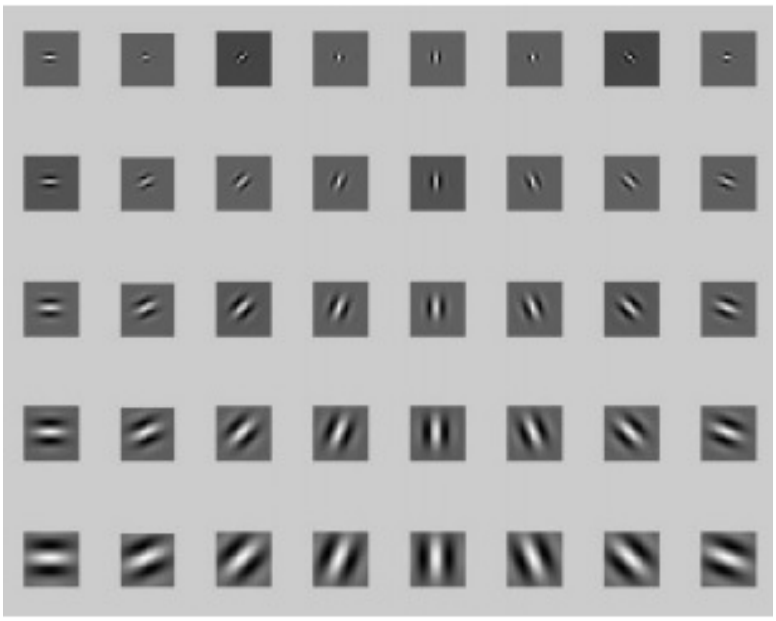


Figure 2.8. Family of 40 Gabor wavelets (From [65]).

2.5.5 Speeded Up Robust Features (SURF)

A hybrid of 2 feature extraction techniques namely speeded up robust features (SURF) and PCA to establish local descriptors [13]. For feature classification they applied K-means algorithm on the clustering of the local descriptors and the global similarities and local similarities are combined in order to classify the images.

The interest point descriptor finds the regions of interest in any given facial image. The descriptor uses an approximate of the Hessian matrix, and the determinant of this matrix represents the blob response at that certain location in an image [13]. An interest point is found when the blob response is a local maximum at that scale and location in an image.

To find the SURF descriptors a square region of the blob response is extracted and orientated in the dominant direction and the region is split up into 4 smaller sub regions. The features at the interest point are found by summing the Haar wavelet response in that region. In the end the SURF descriptor vector has 128 elements. Since it is a hybrid technique PCA is applied to the SURF feature vector in order to reduce complexity with regards to dimensionality and also improve computational efficiency. When PCA is applied to the feature vector it then estimates the projection matrix which is in turn used to compute the new feature space of the hybrid method of SURF and PCA combined [13]. The repeated projecting of the matrix serves a crucial purpose of discarding distortions in the feature vector and preserves interest points.

2.6 FEATURE CLASSIFICATION STAGE

2.6.1 Neural Networks

An artificial neural network (ANN) is one type of a classification technique that is used in various applications in order to predict unknown data. ANNs have nodes that span an artificial network of neurons. Any ANN is composed of three separate kinds of nodes namely, input nodes, hidden nodess and output nodes. Figure 2.9 shows the structure of a single neuron, in this case highlighted in green.

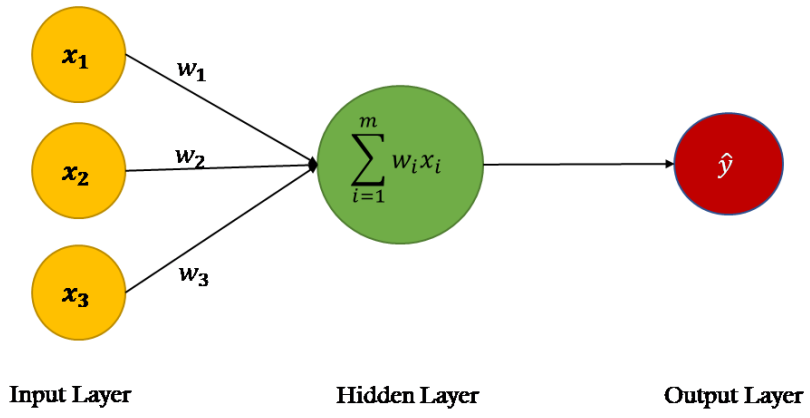


Figure 2.9. Structure of a single neuron.

In the case of Figure 2.9, $x_1, x_2, x_3 \ldots x_n$ represents the inputs that go into the neuron. The inputs vary from being actual observations to some values outputted by one hidden layer neuron to another hidden

layer neuron. The weights of a ANN, denoted by $w_1, w_2, w_3, ...w_n$, represent the strength of connection between two neurons. For example if the weight between node 3 and 4 is high it means node 3 has a greater influence on node 4. The output of a neuron is given by:

$$y = f \cdot (\sum_{i=1}^{m} w_i \cdot x_i). \tag{2.9}$$

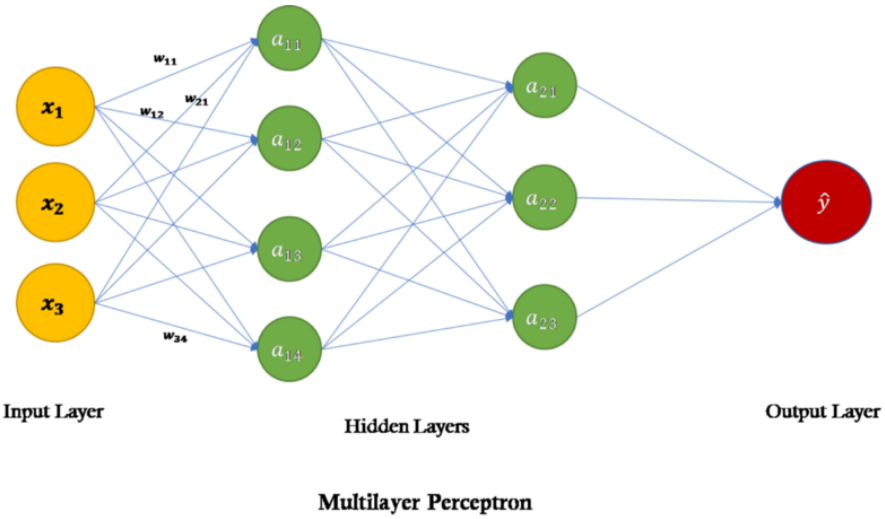


Figure 2.10. Artificial neural network structure

Figure 2.10 shows the typical structure of an ANN with 3 layers. A neural network can have as many layers as are conceivable. The drawback of having many hidden layers is that the complexity of developing them and the run time exponentially increases. For any given NN the input data is a numeric expression, also known as activation values. The higher the numeric expression the higher the activation of that neuron. This information is then passed through the rest of the NN from node to node up to the output layer. During the passing of information, this information changes the weights, transfer functions, excitation and inhibition of the network. When each node receives data it's weights changes accordingly, and utilises the transfer function to modify the data. The result is then summed and passed through the rest of the hidden layers until the output node is reached or a stopping condition is reached.

2.6.1.1 Variants of Neural Networks

A. Feed Forward Neural Networks

A FFNN is a neural network that consists of the three basic layers, namely input layer, hidden layers and the output layer. Information can only travel in 1 direction in a FFNN. Each neuron's calculations are based on the weighed sum of its input supplied to it. Figure 2.11 shows the basic structure of a

FFNN. The calculated sum of one layer becomes the input to the next layer. This process continues until the output layer is reached and a decision is made.

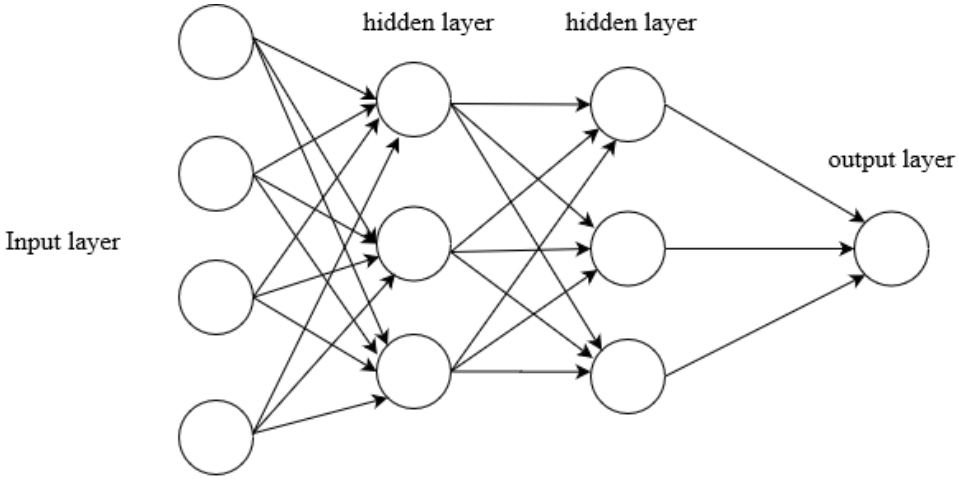


Figure 2.11. Feed forward neural network structure

FFNNs have been used in a hybrid structure where a different method is used for feature extraction and the FFNN is used for feature classification and recognition [67], [68], [69]. Singular value decomposition (SVD) is a statistical model used to reduce dimensionality of input data. It was used to extract features of a human face and then a FFNN was used to do feature classification [67]. Feature based extraction entails using specific features of a face for face recognition while appearance based uses the entire face for recognition purposes [68]. Another hybrid for face recognition was combining feature based and appearance based feature extraction methods to achieve a higher recognition rate. Global information was represented by concatenating the pixel intensities of an image. Local information was obtained by the entropy of the overlapping blocks [68].

PCA is another statistical model often used for feature extraction [69]. It performs the task of reducing the input data from a high dimension to a lower dimension. It achieves this by converting a set of correlated data into uncorrelated data in a lower subspace [55], [69]. The largest principal component onto which data is projected provides the largest variance of the input data and the second orthogonal principal component represent the second largest. The output feature vector was then used as input into a FFNN.

2.6.1.2 Convolutional Neural Networks

A convolutional neural network (CNN) is a neural network that consists of a convolutional layer, a pooling layer, a fully connected layer and then finally the output layer. The convolutional layer also known as the kernel layer is responsible for reducing the dimensionality of input data. This reduction of dimensionality also extracts the low level features of a face for example, the edges, gradient, colour

and orientation [70]. CNNs also contain numerous convolutional layers to extract high level features like the eye region, nose and mouth features. In the case of an image, say with width and height w, h respectively, a square kernel of dimensions $n \cdot n$ where n is sufficiently smaller than both w and h strides the entirety of the image with some set striding value. This kernel would be summing the pixel values in those regions according to some pre-set function.

The following matrix shows a 3 by 3 kernel filter that does matrix multiplication over that potion of the image which the kernel is hovering. The filter can be set to move vertically or horizontally over the entire image, typically it moves left to right until the all of the face image has been completely traversed.

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \tag{2.10}$$

The next layer in a CNN is the pooling layer. In order to reduce the computational power required to extract features in any given image, the pooling layer reduces the spatial size of the convolved features.

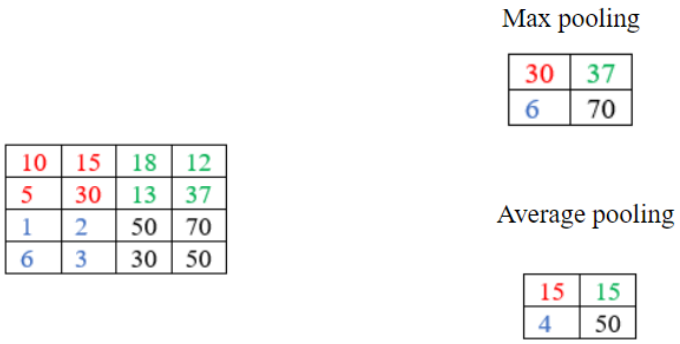


Figure 2.12. 3x3 max pooling and average pooling

Figure 2.12 shows the two types of pooling: max pooling and average pooling. As the name suggests max pooling only takes the highest value in that region while average pooling averages all the values in that region. Max pooling also acts as a noise suppressant as it discards activations that are noisy. After the convolutional layers and pooling layers there is the fully connected (FC) layer. The FC layer holds all the information necessary for classification as it is the feature vector for the input image. It can be fed into other NNs as input for classification.

CNNs have been used in a variety of manners from hybrid approaches on still images to trunk-branch ensemble CNN in video-based face recognition [47], [71–73]. The underlying principle used in conditional convolutional neural networks (c-CNNs) is that different people have different facial features. Thus when input images of different people are fed into a CNN, there are different activations on the same neuron [72]. Traditional CNNs have a fixed width kernel for striding an input picture, adopted a c-CNN approach where the kernels were dynamically activated for different conditions [72]. Consequently, the route in which the kernels were activated were specific in nature to any given face image and they revealed the distribution of underlying modalities [72]. This approach optimised the number of kernels used with any given image. This increased its efficiency by not activating useless kernels in any given sub-window when striding.

Multiple hybrid approaches have been put forward when dealing with CNNs, one such approach is the stacked principal component analysis CNN (SPCA-Net) [73]. SPCANet differs from the traditional CNN in the calculation of the kernel in the convolutional layer. This kernel is found by employing PCA to learn the kernel instead of using the gradient descent method [73]. Instead of using the traditional Sigmoid function and the rectified linear unit (ReLU) in the non-linear layer, a Heaviside function was employed which resulted in binary outputs. The binary outputs are summed with their respective weighed bits [73].

A trunk based ensemble CNN (TBE-CNN) is a neural network that was adapted for video files [47]. This method was applied to video files that contained faces. Artificial blur was applied to videos to simulate real world conditions in which the video taken by a surveillance camera was of poor quality. TBE-CNN is a method which has 1 major truck network and several branches, the major trunk was responsible for learning the holistic face representation while the branches were responsible for learning surrounding patches of the face images where blur was artificially simulated. To speed up computations the middle and lower convolutional layers were shared between the trunk and branches while the high level convolutional layers were optimised independently [47]. The output from these layers were concatenated in order to form a comprehensive face representation. A mean distance regularised triplet loss (MDRTL) function was used to train the TBE-CNN since labelled information is fed into the NN this function took full advantage of that [47]. It was tested on COX Face video database and YouTube Faces databases and it achieved an accuracy of 95.86% and 96.66% respectively [47].

2.6.2 Support Vector Machines

A support vector machine (SVM)is a classification technique that is derived from statistical learning theory [74]. They are superior to other binary feature classification techniques as they can solve non-linear problems with high accuracy. SVMs began as a binary classification method that could only discriminate between two classes [36]. The two classes were only the face class and the non-face class.

They were later adapted to fit multi-class problems which suited face recognition with more than two classes [75], [76].

2.6.3 Decision Trees

Decision trees go a step further as compared to regression models [77]. They make use of the tree structure when classifying input data. This tree structure is successively built by splitting the input data into fewer and fewer entries [77]. When the entire structure has been developed, it contains leaf nodes and decision nodes. A leaf node indicates a classification decision. A decision node has at least two branches stemming from it. These branches indicate separate decisions. The root node is the best predictor for any given input data as it contains all the relevant branches needed to classify any numerical or categorical data [78].

2.6.4 Random Forests

Random forests, also known as random decision trees, they use a technique called ensemble learning when classifying data and doing regression. They differ from the classical random forest in that during the training phase of the algorithm they construct multiple decision trees rather than only a single tree [79], [80]. They output the mean prediction of all the individual trees that were constructed. Random forests are an improvement over decision trees as they do not over fit to the training set.

2.6.5 Naive Bayes Classifier

This classifier is largely based on the Bayes' theorem which assumes linear independence between predictors in a system [81]. The naive classifier makes the logical assumption that the presence of one feature in input data is completely unrelated to the presence of another feature. Whether these two features are related the model makes the assumption of independence, thereby ensuring that each feature contributes independently to the probability [81].

2.6.6 Logistic Regression

Logistic regression is another statistical classifier that analyses input data where there are one or more independent variables that have the ability to detect the outcome of the decision [81]. The outcome is strictly measured with a binary output variable, this variable is only capable of denoting one or the other, and the output cannot be both. The aim of logistic regression is to find a model that fits best the dependent variable to the independent variables [81]. The dependent variable is the outcome variable while the independent variable is the predictor.

2.6.7 Nearest Neighbour

The nearest neighbour algorithm, commonly known as the k-Nearest Neighbour (k-NN) is a supervised learning algorithm. It takes labelled input data split into distinct classes and uses this data to learn new

unknown data [82]. In order to classify a point it assesses the closest k neighbours to that point, and it subsequently classifies that point into a class that is the majority of the closest neighbours [82]

2.7 CONCLUDING REMARKS

In this chapter, a comprehensive literature review of facial recognition techniques, which include pre-processing, feature extraction and feature classification, was presented. The history of face recognition is both complex and long. The early stages of face recognition can be traced back to Ronald Fisher who presented how to classify different flowers. In these early stages, little was known of actually recognising faces. Fisher used a statistical method, which was later known as Fisher's linear discriminant, to accomplish this discrimination.

This chapter also looked at the theory that underpinned different algorithms used for the three key stages of a face recognition system: pre-processing, feature extraction, feature classification and recognition. The pre-processing stage entails removing any noise from images, light variations and consequently normalising images. This is done so as to make recognition uniform and not have noise play a major role in the extraction and classification step. The techniques in image pre-processing, that were looked at, include turning all images to greyscale, histogram equilisation and down-sampling images to a lower resolution so as to speed up processing.

This chapter also looked at what the feature extraction and classifications stages were from a theoretical standpoint. It looked at the different algorithms used to accomplish the goals of feature extraction and classification. For feature extraction, several methods were looked at namely: hidden Markov models, LDA, DCT and Hough transforms. Feature classification techniques that were presented ranged from neural networks, random forests, decision trees, naive Bayes classifiers and decision trees.

CHAPTER 3 PAST AND PRESENT METHODS

3.1 CHAPTER OVERVIEW

The preceding chapter looked at the theoretical side of the three stages a complete face recognition system and some of the algorithms used to accomplish such a goal. In this chapter an in depth review and technical analysis was performed on some of the algorithms that were compared against the proposed weighing and segmenting technique. In Section 3.2 discrete cosine transform is presented as one of four methods that were implemented in order to do comparisons against the proposed technique. In Section 3.3 independent cosine transform is presented. This method decomposes a source signal into a linear combination of unknown variables. In Section 3.4 non-negative matrix factorisation is presented. This method factorises an input image matrix into two separate output matrices used for face recognition. In Section 3.5 sparse representation-based classification is presented. This method represents data as a linear combination of the input elements. Input elements are commonly referred to as atoms that make up a dictionary of atoms. These four methods were implemented so as to control all variables and have the same exact conditions on the databases, computational platform and the pre-processing techniques. This allows there to be a uniform comparisons between all the methods.

3.2 DISCRETE COSINE TRANSFORM

Discrete cosine transform (DCT) is a mathematical model that explicitly transforms a finite sequence of input data into a finite sequence of cosine functions that oscillate at various frequencies [83], [84]. These frequencies are discretely separated and are thus not continuous. It was first put forward in 1974 as a technique for data compression in images. Using cosines rather that sines to approximate an input signal is a crucial form of compression as fewer cosines are required as compared to sine functions [18]. Since then it has become extremely popular as several versions of it have been put forward [83], [85]:

The DCT-I-E

$$X(k) = \sqrt{\frac{2}{N}}\alpha(k)\sum_{n=0}^{N}\alpha(n)x(n)\cos\frac{kn\pi}{N}, \quad k = 0,1,\cdots,N, \tag{3.1}$$

where, $\alpha(k)$ is found by using:

$$\alpha(k) = \begin{cases} \frac{1}{\sqrt{2}} & : k = 0 \\ 1 & : k \neq 0, \end{cases}$$

$$\alpha(n) = \begin{cases} \sqrt{\frac{1}{2}} & : n = 0 \\ 1 & : n \neq 0. \end{cases}$$

The DCT-II-E

$$X(k) = \sum_{n=0}^{N-1}\alpha(n)\,x(n)\cos\frac{(2k+1)n\pi}{2N}, \quad k = 0,1,\cdots,N-1, \tag{3.2}$$

where, $\alpha(n)$ is found by using:

$$\alpha(n) = \begin{cases} \sqrt{\frac{1}{N}} & : n = 0 \\ \sqrt{\frac{2}{N}} & : n \neq 0. \end{cases}$$

The DCT-III-E is also known as the inverse-DCT. If applied to the output of a DCT transform, it can recover the original data, depending on the number of cosine components used [83].

$$X(k) = \alpha(k)\sum_{n=0}^{N-1}\,x(n)\cos\frac{k(2n+1)\pi}{2N}, \quad k = 0,1,\cdots,N-1, \tag{3.3}$$

where, $\alpha(k)$ is found by using:

$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{N}} & : k = 0 \\ \sqrt{\frac{2}{N}} & : k \neq 0. \end{cases}$$

The DCT-IV-E

$$X(k) = \sqrt{\frac{2}{N}}\sum_{n=0}^{N-1}x(n)\,\cos\frac{(2k+1)(2n+1)\pi}{2N}, \quad k = 0,1,\cdots,N-1. \tag{3.4}$$

All of the above DCTs are one-dimensional in nature, this dissertation implemented the two-dimensional DCT.

DCT is fast, accurate and robust face recognition system. The robustness of this system to illumination changes, is increased by employing a normalisation technique. In order to maximise compression, DCT mimics the information packing of the Karhunen-Loeve transform (KLT). DCT is obtained by optimally transforming the input data according to a specific function [84]. On one hand the KLT transform is data-dependant while the DCT is not, thus in certain practical scenarios DCT is much more feasible. It is much more feasible when DCT is used for signal representation due to its information packing ability and computational complexity, this can be attributed to its data independence [84]. When DCT is applied onto an image it separates it into different spectral sub regions of differing importance. Very few coefficients are needed to recover the original image since most of the remaining coefficients lead to redundancy in representing the image [86] [84]. The low frequency coefficients correspond to those with the highest magnitude. These high frequency components relate to illumination variations and the forehead and cheeks region due to their smoothness. The medium frequency coefficients are mostly related to the general shape and structure of a face.They can be found in the medium region of the face. Lastly the highest frequency components correspond to those with the lowest magnitude and they can be found in the lower region of the DCT matrix.

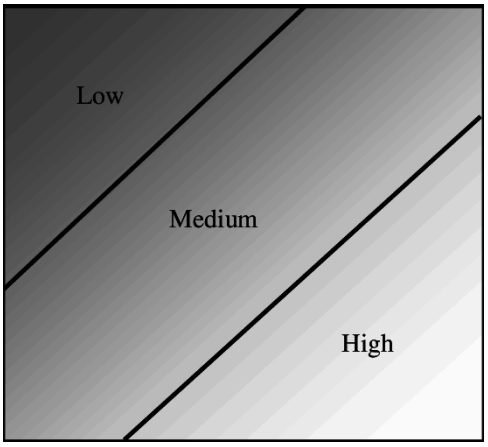


Figure 3.1. DCT coefficient Matrix showing the three regions.

Figure 3.1 shows the general regions of the DCT coefficient matrix. It shows where the frequency components are located. The performance of the DCT is mostly affected by any sort of changes made to the region where low frequency components are found.

3.2.1 Implementation

The DCT implemented can be summed up using 3 sections with various sub-steps. Algorithm 1 summarises the steps that were taken to implement it. The full explanation of what took place follows it.

Algorithm 1 DCT implementation

```
1: procedure DCT)
2:     Load all images into matrix
3:     Convert all images to grayscale
4:     Perform histogram equilisation on all images
5:     for X_{i=1} to N do
6:         Normalise image X_i                          ▷ individual image
7:         Initialise matrix to zeros
8:         Perform block two-dimensional DCT on 8 x 8 blocks
9:         Scale DCT coefficients by the maximum value
10:         Select 1 to 64 coefficients for quantisation
11:         Perform quantisation on the coefficients
12:         Do zigzag scanning to produce the one-dimensional matrix
13:         Encode the Matrix
14:         Discard the high frequency coefficients (compression)
15:         Save resulting compressed coefficients as the feature vector.
16:     for X_{i=1} to N do
17:         Implement nearest neighbour classifier
18:         Evaluate performance metrics (accuracy and error rates)
19:     Plot metrics
```

3.2.1.1 Normalisation

The first step is to load all the images in the database used into the two-dimensional vectors. This was easily achieved by using *numpy* functions using Python, that meant the images were characterised by:

$$I(x,y) = (x_1,y_1),(x_2,y_2),...(x_N,y_M), \tag{3.5}$$

where (x_1,y_1) represents the pixel value at that particular location. M & N were the dimensions of the pictures used. The AR face database and Yale faces databases were used for all experiments, these databases are discussed in Chapter 4 [87], [88], [89]. The matrix where the DCT coefficients would be held was initialised to zeros, naturally the matrix would have the dimensions M & N of the database images.

The next step was to initialise the empty matrix to zeros:

$$\begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

Illumination plays a major role in face recognition. The low frequency components produced by DCT are affected by illumination, thus there is a need for normalisation. In this case, illumination normalisation was done according to the Hummel's histogram modification technique [90]. This was done by choosing a pre-set histogram and then applying a gray scale transformation that would make the input histogram resemble the pre-set one. The images in the databases used were captured over various days and the images contained variations in illumination. The normalisation value was obtained as follows:

$$\text{Normalization Value} = \frac{\text{Average va1ue of registered image}}{\text{Average value of test image}} \tag{3.6}$$

This value was multiplied to each and every pixel in the test image resulting in an image with a relative intensity to that of the registered images.

3.2.1.2 Obtaining the DCT Coefficients

The image was divided into 8 x 8 pixel blocks, and the DCT was applied to each individual block at each iteration. The next step was to perform a block DCT for a given image on 8 x 8 blocks. Two-dimensional DCT was implemented according to Eq. 3.7. Two-dimensional DCT was implemented instead of the one-dimensional because its faster to compute. One-dimensional DCT would have to be applied onto the image twice, once row-wise and then column-wise [91]. This in turn sped up the computational time.

$$F(u,v) = \sqrt{\frac{2}{N}}\sqrt{\frac{2}{M}}\sum_{i=0}^{N-1} A(i)\cdot\cos\left(\frac{u(2i+1)\pi}{2N}\right)\cdot\sum_{j=0}^{M-1}\mathrm{A}(j)\cdot\cos\left(\frac{v(2j+1)\pi}{2M}\right)\cdot f(i,j), \tag{3.7}$$

$$A(i) = \begin{cases} \frac{1}{\sqrt{2}} & : \text{for } u = 0 \\ 1 & : \text{otherwise,} \end{cases} \tag{3.8}$$

$$A(j) = \begin{cases} \frac{1}{\sqrt{2}} & : \text{for } u = 0 \\ 1 & : \text{otherwise,} \end{cases} \tag{3.9}$$

where, $f(i,j)$ is the intensity of the pixel at (i,j), $F(i,j)$ is the DCT coefficient for the pixel (i,j), and M x N are the dimensions of the input image. The images in the AR database were cropped resulting in an image of 120 x 165 pixels. This meant that the number of calculated coefficients were 19800. The original dimensions of images in the Yale faces database was 340 x 243 pixels. This database was scaled to 160 x 120 pixels for uniformity sake and for faster computational time. When DCT was applied to it also resulted in the same number of coefficients as the former database. Compression is achieved since much of the signal energy lies at the lower frequencies and the higher frequencies are

often very small that they can be neglected. Due to the DCTs compression nature only 64 coefficients were taken while matching while the rest were discarded. This is the quantisation step.

3.2.1.3 Zigzag Scanning

The purpose of this step is to order the matrix such that low frequency coefficients accumulate at the top of the matrix while high accumulate at the bottom of the matrix. For each and every 8 x 8 block zigzag scanning was performed. The resulting zigzag was a one-dimensional matrix of size 1 x 64 since a block of 8 x 8 was used.

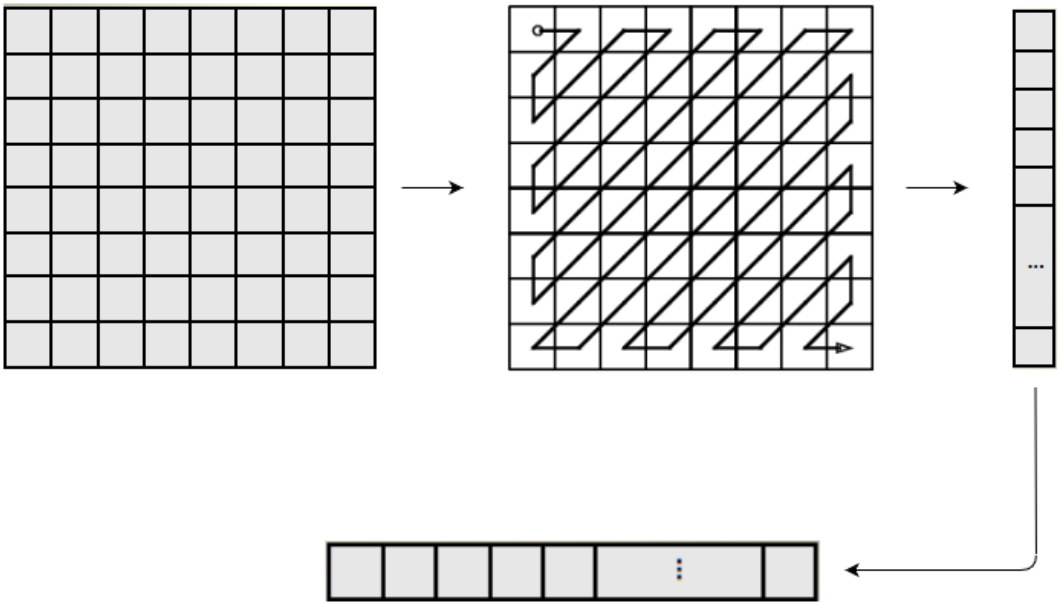


Figure 3.2. Zigzag scanning.

As such the resulting one-dimensional matrix was the feature vector extracted by DCT. This is then what was stored as the feature vector for the images.

3.2.1.4 Euclidean distance measure and recognition

The Euclidean distance measure was used as the classification technique, once the feature vector was extracted. Mathematically speaking it is the straight line distance between two points in a Euclidean space [92]. The resulting norm is called Euclidean norm also known as L^2 Norm [93]. This Euclidean measure can be defined for many dimension, with one-dimensional one described as [92] [93]:

$$\sqrt{(q-p)^2} = |q-p|, \tag{3.10}$$

with p and q as two points on a real line and their distance given by Eq. 3.10.Since the feature vector contained multiple coefficients that were extracted from any given image one-dimensional Euclidean distance measure was not sufficient. Multi-dimensional Euclidean distance measure was used as described by [92] [93]:

$$d(\mathbf{p},\mathbf{q}) = \sqrt{(p_1-q_1)^2+(p_2-q_2)^2+\ldots+(p_i-q_i)^2+\ldots+(p_n-q_n)^2}, \tag{3.11}$$

$$d(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{i=1}^{n}(p_i-q_i)^2}, \tag{3.12}$$

where, d is the distance from a query image to a particular training image, n was the number of dimensions in the feature vector, p was the feature vector of the query image from the test set, and q was the feature vector of a particular image in the training set. For recognition purposes the shortest distance, d was taken as the distance that corresponded to the image that closely resembled the query image. Thus that subject was regarded as the closest match.

3.3 INDEPENDENT COMPONENT ANALYSIS

Independent component analysis (ICA) aims to minimise any dependencies in both the second order and higher order inputs. It was born out of the *blind source separation* (BSS) problem in which a recorded signal is decomposed into a linear combination of unknown variables [94–96]. If **s** is the vector of unknown source signals $s_1, s_2, ..., s_n$, **x** the vector of observed mixtures $x_1, x_2, ..., x_m$ and **A** the unknown mixing matrix [97]. In this case **A** is an m x n matrix matrix. Features are represented by the columns of **A** whilst the coefficients of the "ith" feature in the observed data vector **x** is represented by s_i. This mixing model can be represented as:

$$\mathbf{x} = \mathbf{A}\mathbf{s}. \tag{3.13}$$

In order to visualise Eq. 3.13 consider **A** as a 2 by 2 matrix, (note in this dissertation, the dimensions of **A** depended on the number of pixels and number of total images that were used). Bear in mind that, initially **A** is unknown.

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{pmatrix} 5 & 7 \\ 3 & 12 \end{pmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \Longrightarrow \begin{cases} x_1 = 5s_1 + 7s_2 \\ x_2 = 3s_1 + 13s_2. \end{cases} \tag{3.14}$$

ICA makes a key assumption that all input signals are statistically independent and that **A** is an invertible mixing matrix. The aim of ICA algorithms is to find a **A** such that:

$$\mathbf{u} = \mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{A}\mathbf{s}. \tag{3.15}$$

This model can be represented by Figure 3.3.

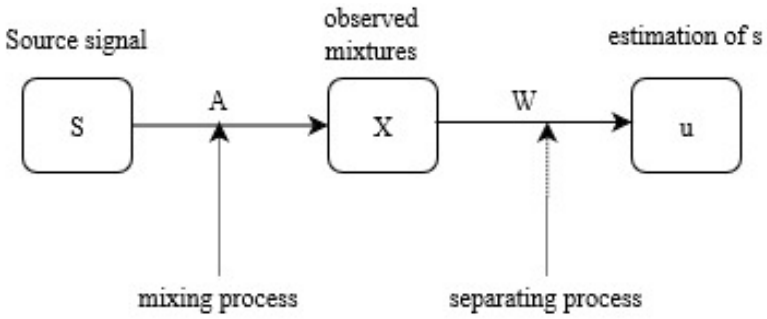


Figure 3.3. The BSS model [97].

3.3.1 Architectures

3.3.1.1 Architecture I

Several researchers have proposed that since architecture I generates spatially localised features when ICA is applied it produce statistically independent basis vectors [97–100]. This leads to better facial action recognition on images with local distortions or partial occlusions due to the influence of small parts of the image on the features.

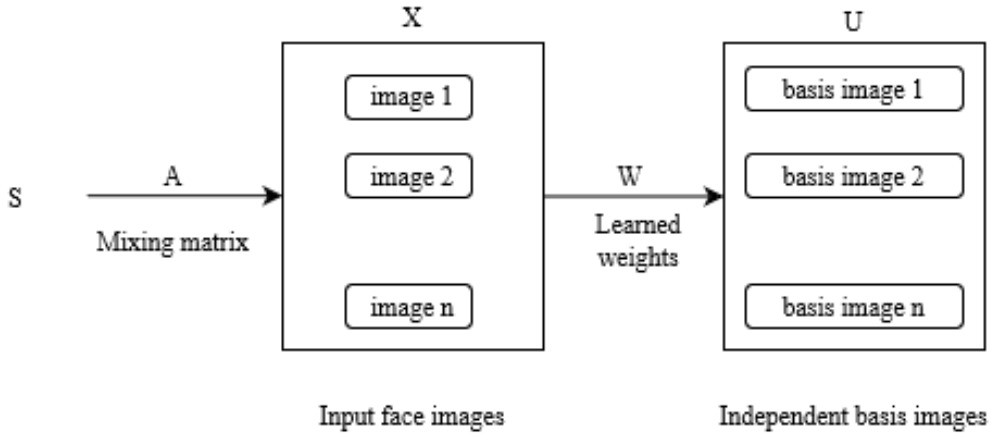


Figure 3.4. Architecture I.

3.3.1.2 Architecture II

Image features in the ICA space can be interpreted as the basis vectors that define any underlying subspace. In architecture II each and every pixel influences the image features there this architecture produces global features [98]. However beacuse of its global nature this architecture is susceptible to distortions and partial occlusions. Regardless of this, it has been shown that architecture II produces better object recognition because of its holistic nature [98], [101]. When this architecture is applied instead of producing spatially independent basis vectors it produces statistically independent compressed images [94].

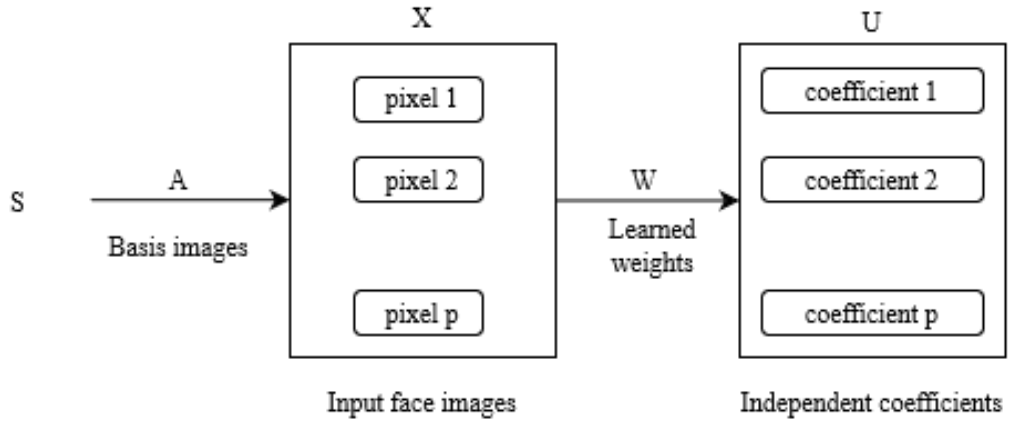


Figure 3.5. Architecture II.

3.3.2 Implementation

There are four algorithms that can be used in order to compute the ICA namely; second order blind identification (SOBI), FastICA also known as Hyvarinen's fixed point algorithm, Infomax and joint approximation diagonalisation of eigenmatrices (JADE) [94], [101–103]. SOBI utilities second-order statistics to decompose the time-correlation of input signals, it mainly computes the whitening, lagged correlation matrices and finally the joint distribution [101]. FastICA utilises the kurtosis for estimating the independent components [103]. The Infomax algorithm is a neural learning method that maximises the entropy and presents a natural gradient [104]. The JADE algorithm also computes the joint distribution and performs whitening but it only does this on $\frac{n}{2}(n+1)$ eigen matrices. Architecture II was adopted and a variant of the ICA algorithm called FastICA was implemented.

3.3.2.1 Pre-processing for ICA

A. Centrering

The ICA algorithm was simplified by centrering the data matrix **x**. this was accomplished by subtracting the mean given by:

$$\mu = E(\mathbf{x}). \tag{3.16}$$

This step essentially turns **s** into a matrix with zero-mean data. When the mixing matrix was approximated, the average vector of **s** and its centred estimations were added together. The average vector was found by:

$$\mathbf{s} = \mathbf{A}^{-1}\mu. \tag{3.17}$$

B. Whitening

When data is referred to as *white* it means that data's components are completely unrelated to each other [105]. Whitening data was another crucial pre-processing step that was adopted. This was done so as to make the data matrix **x** have unity variance. The complexity of the problem at hand was reduced by applying the standard whitening procedure on the data matrix, **x**. This procedure also reduces the dimensionality of the data. The aim of linearly transforming the data is to find components with unity variance, obtained by the following expression:

$$\mathbf{E} = \left\{ \overline{\mathbf{x}} \cdot \overline{\mathbf{x}}^T \right\} = I. \tag{3.18}$$

In simple terms, the new vector $\overline{\mathbf{x}}$'s covariance was equal to the identity matrix, I. Eigen-value decomposition was performed on the covariance matrix as follows:

$$\mathbf{E} = \left\{ \mathbf{x} \cdot \mathbf{x}^T \right\} = \mathbf{V}\mathbf{D}\mathbf{V}^T, \tag{3.19}$$

where, **V** is the orthogonal matrix of eigenvectors of (3.19), and **D** is the eignevalues obtained from the diagonal matrix. These values were simply found by using a numpy function to return only the diagonal values as a matrix **D**.

Inorder to end up with Eq. 3.18 whitening was done by:

$$\overline{\mathbf{x}} = \mathbf{V}\mathbf{D}^{-0.5}\mathbf{V}^T\mathbf{x}. \tag{3.20}$$

The mixing matrix **A** is therefore transformed from having n^2 parameters into the orthogonal matrix $\overline{\mathbf{A}}$ having only $\frac{n(n-1)}{2}$. this was accomplished by:

$$\begin{aligned} \overline{\mathbf{x}} &= (\mathbf{V}\mathbf{D}^{-0.5}\mathbf{V}^T)\mathbf{x}, \\ \overline{\mathbf{x}} &= (\mathbf{V}\mathbf{D}^{-0.5}\mathbf{V}^T)\mathbf{A}\mathbf{s}, \\ \overline{\mathbf{x}} &= \overline{\mathbf{A}}\mathbf{s}. \end{aligned} \tag{3.21}$$

Eq. 3.21 was performed by using the Python numpy library for matrix multiplication. Centrering plus whitening combined is popularly known as sphering and this had the effect of removing the second-order statistics of the data [102]. The variance was equal to 1 and the covariance and mean were 0.

3.3.3 FastICA

- $\mathbf{w}_i$: The variable represented the column vector of the unmixing matrix **W**.
- $\mathbf{w}_i^+$: was used as temporary variable used in the calculation of $\mathbf{w}_i$.
- E(...) is the mean of the respective data. It is also known as the expected value.

Algorithm 2 FastICA procedure

1: **procedure** FASTICA($\mathbf{X}$, N)
2: **for** 1 to N **do**
3: Initialise the weight vector w randomly
4: Get the sub index $\mathbf{w}_i$
5: $\mathbf{w}_i^+ = E(\phi'(\mathbf{w}_i^T\mathbf{X}))\mathbf{w}_i - E(\mathbf{x}\phi(\mathbf{w}_i^T\mathbf{X}))$
6: $\mathbf{w}_i = \frac{\mathbf{w}_i^+}{||\mathbf{w}_i^+||}$
7: **if** i equal to 1 **then**
8: goto step 4
9: **else**
10: continue
11: $\mathbf{w}_i^+ = \mathbf{w}_i - \sum_{j=1}^{i-1} \mathbf{w}_i^T\mathbf{w}_j\mathbf{w}_j$
12: $\mathbf{w}_i = \frac{\mathbf{w}_i^+}{||\mathbf{w}_i^+||}$
13: **if** not converged **then**
14: goto step 5
15: **else**
16: **if** all components extracted **then**
17: END
18: **else**
19: $i = i + 1$
20: goto Step 4
21: output $\mathbf{W} \leftarrow [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N]$
22: output $\mathbf{S} \leftarrow [\mathbf{W}^T\mathbf{X}]$

- $\phi'(\ldots)$ is the derivative of $\phi(\ldots)$

The stopping condition was made so that once a given $\mathbf{w}_i$ had converged the next $\mathbf{w}_{i+1}$ was made orthogonal to the last one according to step 11 and 12 in algorithm 1.

The images were arranged in such a way that each row of the matrix $\mathbf{X}$ was a different image. In order to accurately depict architecture II $\mathbf{X}$ is transposed so that images were represented as column vectors according to $\mathbf{X} = \mathbf{X}^T$. Pixels were the random variables and the images were the observations or trials [102]. Intuitively speaking, random pixels a and b when moving across one image are statistically independent since knowing pixel a has no bearing on knowing pixel b. Source separation was performed on each and every pixel. An image was represented by the rows of the learned matrix $\mathbf{W}$. While $\mathbf{A} = \mathbf{W}^{-1}$ contained the basis images.

Figure 3.6 shows three randomly extracted basis images from the recovered $\mathbf{A}$ vector. Each individual image was represented as a column in $\mathbf{A}$.

During the training phase of FastICA two databases were used for both testing and training purposes. The databases were split into a $0.7 : 0.3$ ratio between training and testing sets. These training images

Figure 3.6. Three randomly selected basis images.

were the input to the subspace projection and eventually formed the set of basis vectors. Similarly, the testing images were projected to the same subspace using the techniques outlined in this section, and the nearest neighbour method was implemented to find the closest match.

3.4 NON-NEGATIVE MATRIX FACTORISATION

Non-negative matrix factorisation (NMF) is a novel machine learning technique that is part of the unsupervised learning category. It has a wide range of uses such as signal processing, text recognition, face recognition and topic modelling [106]. Strictly speaking NMF is where an input matrix is factorised into two separate output matrices [107–110]. The output matrices are constrained so that all their elements are non-negative. NMF is quite easy to grasp but the very concept of factorisation is rather difficult owing to the fact that the initial conditions of the output matrices are difficult to 'guess'.

Let a single image be denoted by $\mathbf{I}(x,y)$, $x \times y$ being the rows and columns that hold the pixel images.

$$I(x,y) = \begin{bmatrix} x_1y_1 & x_1y_2 & \dots & x_1y_m \\ x_2y_1 & x_2y_2 & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_ny_1 & x_ny_2 & \dots & x_ny_m \end{bmatrix} \tag{3.22}$$

Initially the image is flattened so that it is a one-dimensional matrix with one column and $n \times m$ rows. Classes are represented as $\mathbf{C}$:

$$\mathbf{C} = \{C_1, C_2, C_3, \dots, C_g\}, \tag{3.23}$$

where g is the number of classes. Each and every class was limited to a certain number of image. The

Yale faces database [89] had ten images per class while the AR database [87] was limited to 20 images per class and finally 15 subjects from the ORL database were used [88]. The number of images in a class is denoted by z.

NMF aims to factorise a matrix $\mathbf{X}$ such that:

$$\mathbf{X}_{pn} \approx (\mathbf{W} \cdot \mathbf{H})_{p \cdot n} = \sum_{a=1}^{r} \mathbf{W}_{p \cdot r} \mathbf{H}_{r \cdot n} = \hat{\mathbf{X}}. \tag{3.24}$$

$\hat{\mathbf{X}}$ is the approximation of the original matrix. Since there is no 100% guarantee that by simply guessing the coefficients of both matrix $\mathbf{W}$ and $\mathbf{H}$ would lead to an original reconstruction of $\mathbf{X}$ approximation is the best alternative in this situation [111], [112].

Depending on the number of classes in any given database, all the images are concatenated to make the $\mathbf{X}$ matrix with $n \times m = p$ rows and $g \times z = n$ columns.

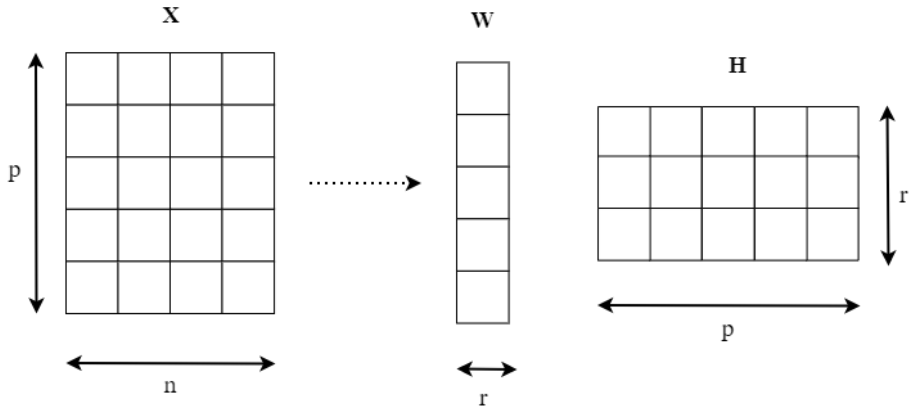


Figure 3.7. Matrix visualisation.

Each column in the image database $\mathbf{X}$, contain the non-negative pixels that makes up an image which are regarded as the dimensions. The number of rows are the total number of images in the database, they can be regarded as the n-dimensional data points in a linear subspace [107]. $\mathbf{X}^{+}$ being the approximation of the original dataset. Each column of the matrix $\mathbf{W} \in \mathbb{R}^{p \times r}$ such that each column is $\mathbf{W}(;,k) = w_k$.The columns of $\mathbf{H}$, represents the horizontal and vertical position of $\mathbf{X}(;,j)$ in the basis $\mathbf{W}$ symbolised by $\mathbf{H}(;,j) = h_j$ [113].

The matrices $\mathbf{W}$ and $\mathbf{H}$ can be found iteratively, this is done by multiplying the current value of the approximation by some factor that leads to an improvement and consequently convergence to some final value. this final value converges to the corresponding value in the original dataset. This is accomplished by minimising $||X - WH||^2$. In order to approximate 3.24 the quality of approximation

has to be quantified according to some cost function [114]. The distance between two non-negative matrices A and B can be found by multiple equations. The first one being the simple Euclidean distance measure between A and B:

$$||X - WH||^2 = \sum_{ij}(X_{ij} - (WH)_{ij})^2, \tag{3.25}$$

where the lower bound is equal to 0 subject to $A = B$.

The second measure is a divergence based on the Kullback-Liebler divegence [114]:

$$D(V||WH) = \sum_{ij}\Big(X_{ij} \times \log\frac{X_{ij}}{(WH)_{ij}} - X_{ij} + (WH)_{ij}\Big). \tag{3.26}$$

3.4.1 Gabor Kernel

A Gabor filter is a linear filter used in image processing for texture analysis. It analyses whether in any given local patch of an image, there are frequencies that are oriented in a specific direction [115], [116]. These orientations and frequencies are similar to what is perceived by the human visual cortex [117]. A Gabor filter is defined by a function that takes seven parameters as input [107], [118]:

$$g((x,y),\lambda,\theta,\psi,\sigma,\gamma) = \exp(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}) \cdot \exp(i \cdot (2\pi\frac{x'}{\lambda} + \psi). \tag{3.27}$$

The real part can be found by [118]:

$$g((x,y),\lambda,\theta,\psi,\sigma,\gamma) = \exp(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}) \cdot \cos((2\pi\frac{x'}{\lambda} + \psi). \tag{3.28}$$

The imaginary part can be found by [118]:

$$g((x,y),\lambda,\theta,\psi,\sigma,\gamma) = \exp(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}) \cdot \sin((2\pi\frac{x'}{\lambda} + \psi). \tag{3.29}$$

where, **(x,y)** is the input image. **Lambda,** λ, is the wavelength of the sinusoidal factor that governs how thick the Gabor function is. If it is increased so does the minor radius of the eliptical function. **Theta,** θ, represents the normal's orientation to the Gabor function's stripes. Assigning 0° corresponds to a vertical position of the function. **Psi,** ψ, is the phase offset. **Sigma,** σ, is the standard deviation of the Gaussian envelope. When it is increased this allows more stripes to be present. Whilst when decreased, the envelope tightens reducing the number of stripes. **Gamma,** γ, is the spatial aspect ratio that controls the height of the function, when γ is very small the height is large, for high values of γ the height becomes quite small.

The approximations x' and y' can be found by [107]:

$$x^{'} = x\,\mathrm{Cos}\theta + y\,\mathrm{Sin}\theta, \tag{3.30}$$

$$y^{'} = -x\,\mathrm{Cos}\theta + y\,\mathrm{Sin}\theta. \tag{3.31}$$

In this dissertation a Gabor kernel at five scales was used, and at eight different orientations. The rest of the parameters were:

$$\psi = 0, \tag{3.32}$$

$$\sigma = \frac{\lambda}{2}, \tag{3.33}$$

$$\gamma = 1, \tag{3.34}$$

$$\lambda = [1,2,3,4,5], \tag{3.35}$$

$$\theta = [0^{\circ},45^{\circ},90^{\circ},135^{\circ},135^{\circ},180^{\circ},225^{\circ},270^{\circ},315^{\circ}]. \tag{3.36}$$

Using the above methodology Fig 3.8 shows the first ten Gabor filters that were used in convolving the input image and the filters.

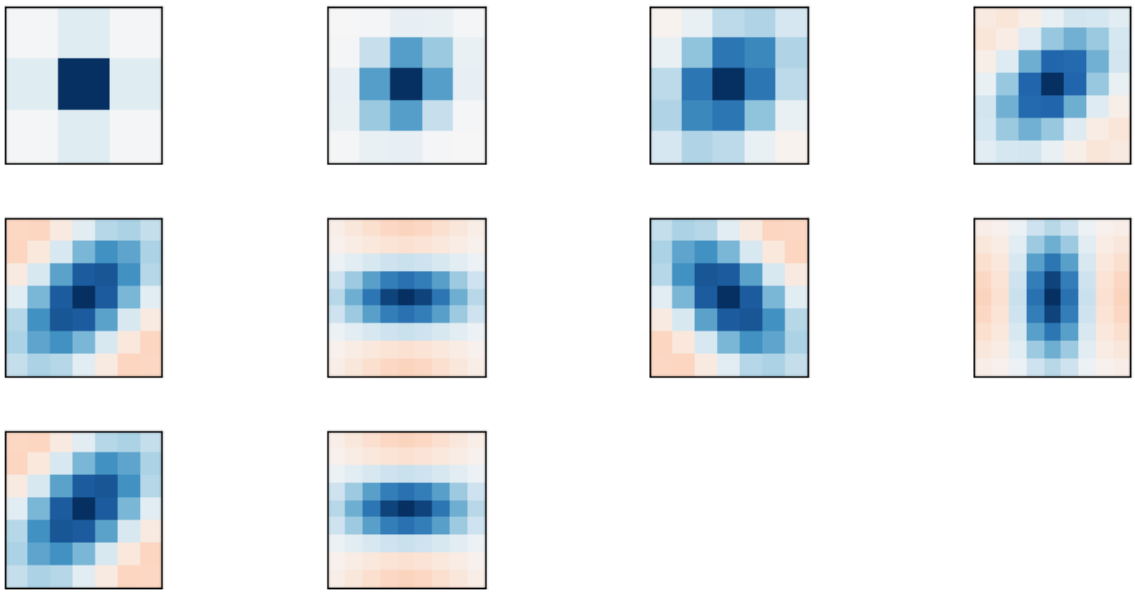


Figure 3.8. The first 10 filters.

Gabor kernels are good feature detectors. They are capable of detecting features such as texture and edges. They are sensitive to edges that have specific orientations at specific locations because of the way they model input data and their intrinsic shape [119], [120].

3.4.2 Implementation

The Gabor kernel was a matrix that contained 40 individual Gabor filters. The 40 filters were obtained by varying and θ according to the following constraints:

$$\lambda = [1,2,3,4,5], \tag{3.37}$$

$$\theta = [0^\circ, 45^\circ, 90^\circ, 135^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ]. \tag{3.38}$$

Figure 3.9 shows the results of convolving an image with two different Gabor filters. Several papers have cited that using 40 Gabor kernels resulted in the best results [116–118], thus this dissertation also adopted the same standard. All the images in the database were convolved with the Gabor kernels resulting in images similar to that in Figure 3.9. These images were then the input to the NMF algorithm.

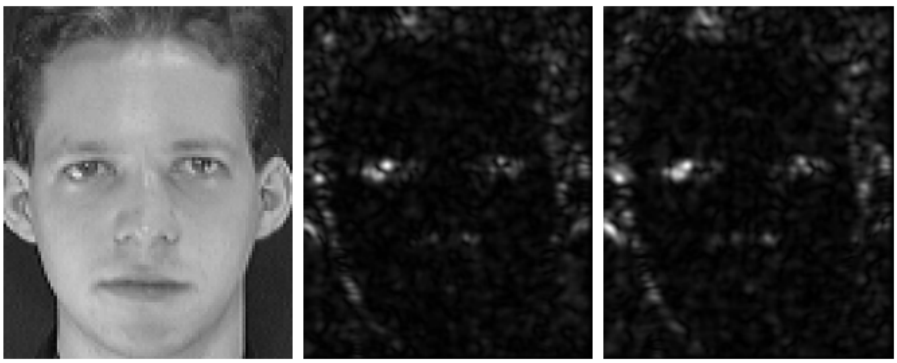


Figure 3.9. Results from convolving subject 1 from ORL database with Gabor filters of different σ and π .

Figure 3.9 shows the original image, followed by an image convolved by a Gabor kernel with $\sigma = 3$ and $\theta = \pi/4$. The last image was the result of a Gabor kernel with $\sigma = 3$ and $\theta = 3\pi/4$.

Algorithm 3 was used to extract features of any given image by convolving different Gabor filters to find feature vectors of different people. This feature vector became the input of the NMF algorithm.

Algorithm 3 Gabor feature extraction

```
1:  procedure GFE
2:      Construct empty F_vector                          ▷ Database feature vector
3:      for i=1 to N_img do                               ▷ total number of images
4:          for i=1 to f_n do                             ▷ frequencies in λ
5:              for j=1 to O_n do                         ▷ orientations in θ
6:                  Construct Gabor filter g((x,y), λ_j, θ_i, ψ, σ, γ)
7:                  Convolve I_i with the Gabor filter g((x,y), λ_j, θ_i, ψ, σ, γ)
8:              end
9:          end
10:     Append feature vector, f_i, of person i to F_vector.
11:     end
12:     Store entire database of extracted features F_vector.
13:     output ← F_vector
```

The next step was to take the output of the convolved images flatten them into one-dimensional arrays and then concatenating these arrays to make two-dimensional database matrix, this matrix became the **X** matrix.

The important step in NMF is to approximate the matrices **W** and **H** known as the weight matrix and feature matrix. When these two matrices are multiplied together, using the dot product rule, its tantamount to finding the sum of the latent features by the weights of each basis image in the original matrix [107]. This is an *NP*-hard problem that can be solved in multiple ways such as [107], [112], [114];

- Multiplicative update
- Alternating Least squares
- Graph Regularised NMF
- Probabilistic NMF
- Kernel NMF

In this dissertation the multiplicative update was used, this technique aims to minimise the error between the approximated dot product of **W**·**H** and the original data **X**. To calculate the error the Frobeus norm was utilised at it is efficient in finding errors between martrices [121]. Once the error has been found the gradient can be found so that the direction in which to go in order to minimise errors can be calculated. Ideally, this should be done until the error becomes 0, nut in real-world scenarios this would be highly time complex thus a certain threshold has to be set.

Algorithm 4 Multiplicative Update

1: **procedure** MUP
2: Randomly initialise **W** and **H** with small random numbers
3: **for** i=1 to S_{max} **do** ▷ S_{max} is maximum step size
4: **for** j=0 to X_{row} **do** ▷ rows of **X**
5: **for** k=0 X_{col} **do** ▷ columns of **X**
6: **if** $\mathbf{X}_{jk} > 0$ **then**
7: Compute error of element.
8: Compute gradient of error.
9: Update **W** and **H**
10: **end for**
11: Compute total error
12: **if** $e < \varepsilon$ **then** ▷ Threshold error, ε, was set at 2^{-8}
13: break
14: **end for**
15: output $\leftarrow$ **W** and $\mathbf{H}^T$ ▷ approximated **W** and **H**

The values of **W** and **H** were found iteratively using Algorithm 4. There are a few needed calculations that are essential to find the right **W** and **H**. The predicted basis image and weight (dot product of **W**· **H**) was found using:

$$\hat{x}_{ij} = w_i^T h_j = \sum_{r=1}^{r} w_{ir} h_{rj}. \tag{3.39}$$

The squared error (Step 7 of Algorithm 4) was computed using:

$$e_{ij}^2 = \left(x_{ij} - \hat{x}_{ij}\right)^2 = \left(x_{ij} - \sum_{r=1}^{r} w_{ir} h_{rj}\right). \tag{3.40}$$

To find the gradient (Step 8 of Algorithm 4) the differential of each element was found by:

$$\frac{\partial}{\partial w_{ir}} e_{ij}^2 = -2(x_{ij} - \hat{x}_{ij}) h_{rj} = -2 e_{ij} h_{rj}, \tag{3.41}$$

$$\frac{\partial}{\partial h_{ir}} e_{ij}^2 = -2(x_{ij} - \hat{x}_{ij}) w_{rj} = -2 e_{ij} h_{ir}. \tag{3.42}$$

In order to avoid skipping over the correct elements of **W** and **H** a learning rate was introduced so as to avoid overshooting from the correct values. The values of **W** and **H** were updated using the learning rate, α. Step 9 of Algorithm 4 was accomplished by:

$$w_{ir}^{'} = w_{ir} + \alpha \frac{\partial}{\partial w_{ir}} e_{ij}^2 = w_{ir} + 2\alpha e_{ij} h_{rj}, \tag{3.43}$$

$$h_{ir}^{'} = h_{ir} + \alpha \frac{\partial}{\partial h_{ir}} e_{ij}^2 = h_{ir} + 2\alpha e_{ij} w_{rj}. \tag{3.44}$$

Convergence was reached once the sum of all the errors tallied to ε which was set at 2^{-8} or the maximum number of iterations had been reached, whichever one came first. The maximum number of iterations was set at 1000. Since the error decreases the slowest when the calculated values were approximating the values of $\mathbf{X}$, it is essential to cap the maximum number of iterations. The convergence criterion was found by when the errors reached a certain threshold obtained by:

$$\varepsilon = \sum_{i=1}^{p}\sum_{j=1}^{n}[X_{ij} * \log(W \cdot H)_{ij} - (W \cdot H)_{ij}]. \tag{3.45}$$

Note: The approximation of the original data is denoted by $\hat{\mathbf{X}}$.
When convergence had been reached it meant that the approximation of the original data $\mathbf{X}$ in Eq. 3.24 had been found. The process of finding $\hat{\mathbf{X}}$ is the training step for NMF. The matrix M_{train} was created and contained the approximation $\hat{\mathbf{X}}$. Testing was performed on three databases.

A split of 0.7:0.3 was used between training and testing samples. The same procedure used to find M_{train} was repeated in order to find the feature matrix, M_{test} of the testing sample. In order to do recognition matrix M_{train} and M_{test} were compared using the Frobenius norm [121]. Euclidean distance was not used as it pertains to one-dimensional data and is only an L^1-norm. Frobenius norm is used for two-dimensional feature vectors, sometimes called L^2-norm. It was done using:

$$||\mathbf{A}||_F \equiv \sqrt{\sum_{i=1}^{m}\sum_{i=j}^{n}|a_{ij}|^2}, \tag{3.46}$$

which was extended to two matrices, let matrix $\mathbf{A}$ represent M_{train} and matrix $\mathbf{B}$ represent M_{test} therefore the similarity measure was calculated using:

$$d_2(\mathbf{A},\mathbf{B}) = \sqrt{\sum_{i=1}^{m}\sum_{i=j}^{n}\left(a_{ij} - b_{ij}\right)^2}. \tag{3.47}$$

This gave the measure of the distance between the two matrices. The matrix with the shortest distance was chosen as the most similar one and thus recognition was achieved. This was done to all training images and the results are captured in Chapter 5, where they are compared to the proposed image segmentation and weighing technique.

3.5 SPARSE REPRESENTATION-BASED CLASSIFICATION

Sparse representation-based classification (SRC) is a machine learning technique that's used to obtain a sparce representation of input data. This data is represented as a linear combination of the input elements [122–124]. Input elements are commonly referred to as atoms that make up a dictionary of atoms. PCA projects data into a lower subspace and looks for a direction that maximises variance between orthogonal planes. On the other hand, SRC atoms are not constrained by the orthogonality

principle [122], [125]. This therefore means that input data need not be reduced in dimensionality. In fact, input data can be projected to a higher dimensional subspace. This seems counter-intuitive at first, but because of the sparse nature of the projected data, the resulting clusters of any given class would be far enough apart to perform efficient classification [122], [126], [127].

SRC has been successfully different fields such as signal recovery, image processing and compressed sensing. In the latter case SRC has been used to recover the original signal with minimal linear measurements provided that the data projected into a higher dimension was sparse enough [122]. Wavelet transform can be applied if the projected data does not meet the sparsity criterion. In signal processing there is a need to infer the original dictionary using as minimal components as possible. The underlying principle is that the original dictionary has to be recovered from the input data, regardless of the fact that it may be corrupt. Being corrupt also means that the input images contain noise, occlusions, varying illumination, that lead to degraded performance due to loss of original discriminatory information. Fourier and Wavelet transforms were used to code and learn the predefined dictionary that was trained to fit the input data [51]. This has the advantage of increasing the sparsity of the projected data leading to this method being used in compression, denoising, decompression, audio and video processing [124], [127].

Say the input dataset, $\mathbf{Y}$, is a set of N original training samples which belong to N classes, the aim is to find a dictionary ,$\mathbf{D}$, and a unique representation ,$\hat{\alpha}$, in such away that $||\mathbf{Y}-\mathbf{D}\alpha||_F^2$ is minimised under the sparsity constraint of r_i [122], [128].

$$\mathbf{Y} = [y_1, y_2, y_3, \ldots, y_k], \quad y_i \in \mathbf{R}^d, \tag{3.48}$$

$$\mathbf{D} \in \mathbf{R}^{d\times n} : \mathbf{D} = [d_1, d_2, d_3, \ldots, d_n], \tag{3.49}$$

$$\hat{\alpha} = [\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n], \quad \alpha_i \in \mathbf{R}^n. \tag{3.50}$$

Consequently, this becomes an optimisation problem [122]:

$$\text{argmin} \sum_{i=1}^{K} ||y_i - \mathbf{D}\alpha_i||_2^2 \quad + \quad \lambda||\alpha_i||_0, \tag{3.51}$$

where the dictionary is defined as follows:

$$C \equiv \left\{\mathbf{D} \in \mathbf{R}^{d\times n} : \quad ||d_i||_2 \leq 1 \quad \mathbf{A}_i = 1,2,3,\ldots,n\right\}, \quad \lambda > 0. \tag{3.52}$$

The dictionary $\mathbf{D}$ can either be under complete or over complete. Under completeness is denoted when the condition $n < d$ is satisfied and over completeness when $n > d$ is satisfied. PCA, FLD, and SVM [28], [51], [129], are based on the former assumption while SRC is based on the latter assumption.

The former assumption represents input data in a lower dimensional subspace which require that data be orthogonal.

3.5.1 Algorithms to find the sparse coding

Sparse approximation, sometimes popularly known as sparse coding, is the procedure of finding optimal sparse coding, $\hat{\alpha}$ given a certain dictionary of atoms, $\mathbf{D}$. There exist a number of algorithms that are used to find this approximation.

3.5.1.1 Method of optimal direction

The method of optimal direction (MOD) aims to solve the minimisation problem when given a limited number of non-zero elements of a representation vector [20]:

$$\min\{||\mathbf{Y}-\mathbf{D}\alpha||_F^2\} \quad \text{subject to} \quad \mathbf{A}_i||\alpha_i||_0 \leq \mathbf{T}, \tag{3.53}$$

where $\mathbf{F}$ indicates the Frobenius norm. This method iterates between updating the dictionary and matching pursuit which are accomplished by computing the analytical solution [20]. $\mathbf{D}$ is then normalised after each and every update to fit the constraints of the sparse coding. Since MOD takes very few iterations to converge it is rather efficient.

3.5.1.2 K-Singular value decomposition

K-Singular value decomposition (KSVD) is a method that updates the elements of the dictionary one by one by performing SVD [130]. It has an absolute upper bound on the number of elements that are used to encode the input data y_i. In this case the upper bound is $\mathbf{T}_0$. This algorithm first fixes $\mathbf{D}$ and find the approximation of $\hat{\alpha}$, once complete it used the following formula to iteratively find the elements of $\mathbf{D}$ [130]:

$$||\mathbf{Y}-\mathbf{D}\alpha||_F^2 \quad = \quad \left|\mathbf{Y}-\sum_{i=1}^{K} d_i\alpha_i^T\right|_F^2 \quad = \quad ||\mathbf{E}_k - d_k\alpha_T^k||_F^2. \tag{3.54}$$

3.5.1.3 Stochastic gradient method descent

This method iteratively does stochastic gradient descent when updating the dictionary and projects it on the constraint set $\mathbf{C}$ [131]. This process is described by:

$$\mathbf{D}_i = \text{proj}_C\left\{\mathbf{D}_{i-1} \quad - \quad \delta_i\mathbf{A}_D\sum_{i\in S}||y_i-\mathbf{D}\alpha_i||_2^2 \quad + \quad \lambda||\alpha_i||_1\right\}, \tag{3.55}$$

where $\mathbf{S}$ is a random subset of $\{1,2,\ldots,\mathbf{K}\}$ and δ_i is a gradient descent [131].

3.5.1.4 Online dictionary learning LASSO

In this approach the optimisation problem becomes [130]:

$$\min \left\{ ||\alpha_i||_1 \right\} \quad \text{subject to} \quad ||\mathbf{Y} - \mathbf{D}\alpha||_F^2 < \varepsilon. \tag{3.56}$$

The estimated values of r_i are found by minimising the least squared error ε when reconstructing the LASSO.

3.5.2 Implementation

It has been extensively shown that the human face when represented as an image cast under various conditions such as; occlusions, light variations, extreme pose, and different emotions lie in a very low dimensional space. The SRC approach takes advantage of this fact by sparse-coding all query images while utilising an over complete dictionary that contains training samples [122], [130]. If the samples of the i-th person, A_i, are sufficiently enough, they can be represented as:

$$A_i = [v_{i,1}, v_{i,2}, v_{i,3}, \ldots, v_{i,n}] \qquad \in R^{m \times n}. \tag{3.57}$$

This SRC approach makes the assertion that, a query input sample was represented by $\mathbf{y}$ ends up constructing a subspace that is linear. So for this example any query image that belonged to class i was effectively represented as a linear combination of samples from that i class. The input sample was represented as:

$$\mathbf{y} = \alpha_{i,1} v_{i,1}, \alpha_{i,2} v_{i,2}, \alpha_{i,2} v_{i,3}, \ldots, \alpha_{i,n} v_{i,n}. \tag{3.58}$$

Algorithm 5 details a high level implementation of the SRC approach. Since the input data is a two-dimensional matrix that contains rows and columns of vector, the first step was to normalise these vectors so that they have unit ℓ_2-norm. This is a common pre-processing technique in machine learning. Vectors are normalised before they are passed on to the next stage of a machine learning algorithm such as SRC and support vector machines (SVM). ℓ_2-norm refers to the act of applying normalisation to a vector so that it can have unit length across its rows [128]. In this case, ℓ_2-norm was applied to the rows of $\mathbf{D}$:

$$\mu = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij}, \tag{3.59}$$

$$\sigma^2 = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(d_{ij} - \mu \right)^2, \tag{3.60}$$

$$\hat{d_{ij}} = \frac{d_{ij} - \mu}{\sqrt{\sigma^2}}, \tag{3.61}$$

where ij refer to the element denoted by position i and j. $\hat{x}$ becomes the new normalised value. Eq. 3.59 Eq. 3.61 show how step 4 in Algorithm 5 was accomplished.

Algorithm 5 Sparse representation based classification

1: **procedure** SRC(**D**,**y**)
2: **Input:** $\mathbf{D} = [d_1, d_2, d_3, \ldots, d_n] \in R^{m \times n}$ ▷ Training samples
3: **Input:** **y** ▷ Test sample
4: Normalise rows of **D** so they have unit ℓ_2-norm
5: Solve minimisation problem so as to code y over d
6: Solve the ℓ_1-minimisation problem using OMP:
7:
$$min||\mathbf{y} - \mathbf{D}\alpha||_2 \quad + \quad \lambda||\alpha||_1 \tag{3.62}$$
8: Compute residuals
9: **for** $i = 1$ to K **do** ▷ limited to K-ranks
10: $e(j)_i = ||\mathbf{y} - \mathbf{D}_i\alpha_i||_2$
11: $E_i = \mathbf{U}\mathbf{G}\mathbf{V}^T$ ▷ apply K-SVD
12: Update d_i using $\mathbf{U}_{:,1}$ ▷ How the dictionary is updated
13: Update $\alpha_{:,1}$ using $\mathbf{V}_{:,1} \times \mathbf{G}_{1,1}$ ▷ How the sparse codes are updated
14: **end for**
15: identity(**y**) = argmin$_i\{e_i\}$
16: **output** ← identity(**y**)

The difficult step was projecting **y** over **D** so as to solve the minimisation problem. This problem was solved using the orthogonal matching pursuit (OMP) algorithm [132], [133]. The least square problem was solved using vector manipulations and systems of equations to find an estimate of the original signal. This estimate was found by solving $\mathbf{y} = \mathbf{D} \cdot \hat{\alpha}$ and inverting **D** [132]. In this instance, **D** was not a square matrix and its inversion was found by using:

$$\hat{\mathbf{D}} = \left(\mathbf{D}^T \cdot \mathbf{D}\right)^{-1} \cdot \mathbf{D}^T. \tag{3.63}$$

The solution of the original equation,$\mathbf{y} = \mathbf{D} \cdot \alpha$, was then computed using:

$$\hat{\alpha} = \left(\mathbf{D}^T \cdot \mathbf{D}\right)^{-1} \cdot \mathbf{D}^T \cdot \mathbf{y}. \tag{3.64}$$

This was accomplished using matrix inversion and multiplication [132]. K-SVD algorithm for Dictionary learning differs from the traditional SVD in the sense that it enforces the K-means algorithm. This limits the number of elements required to approximate an entry in the dictionary, **D**.

Applying the K-SVD to E_i yielded tree decomposed matrices which were used to update elements in the dictionary, **D**, and the sparse representation vector, $\hat{\alpha}$. K-SVD updates the elements of the dictionary one by one by performing SVD [130]. The minimisation problem is reduced to solving

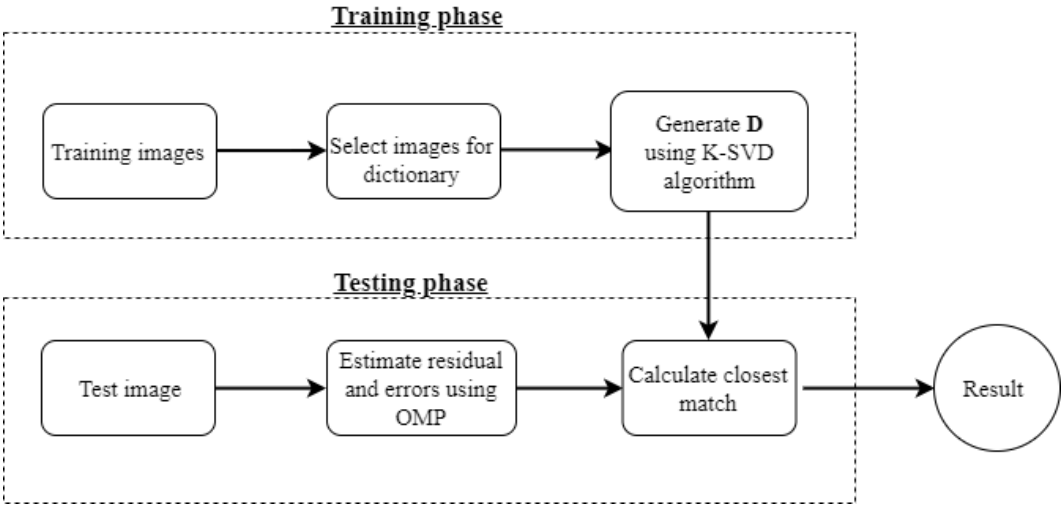


Figure 3.10. Testing and training phase block diagram for the sparse representation-based classification..

$||E_k - d_k\alpha_T^k||_F^2$ as represented in Eq. 3.65 [122].

$$||\mathbf{Y} - \mathbf{D}\hat{\alpha}||_F^2 \quad = \quad \left|\mathbf{Y} - \sum_{i=1}^{K} d_i\alpha_i^T\right|_F^2 \quad = \quad ||E_k - d_k\alpha_k^T||_F^2, \tag{3.65}$$

where α_i represents the i-th row of $\hat{\alpha}$, and d_k is the k-th row of $\mathbf{D}$ [134]. In order to find the solution to d_k, e_k was decomposed into $\mathbf{UGV}_T$ using K-SVD. The first column of $\mathbf{U}$, denoted by $U_{:,1}$, was the solution of d_k and the first column of $\mathbf{V}_{:,1} \times \mathbf{G}_{1,1}$ represented the coefficient vector, α_k [134].

Initially, the correct class was not known and a dictionary was constructed according to: $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_c] = [v_{1,1}, v_{1,2}, \ldots, v_{c,n}] \in R^{m \times N}$, where N is the total number of training samples [122]. The query input image is thus represented as a linear combination of all training images:

$$\mathbf{y} = \mathbf{D} \cdot x_0 \qquad x_0 = [0, \ldots, \alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,6}, \ldots, \alpha_{i,n}, 0, 0, \ldots, 0]^T. \tag{3.66}$$

The vector x_0 refers to all the coefficients which are zero except for those that belong to the correct class i. So once all the coefficients, $\alpha_0(j)$, that belong to the j-th class are extracted the approximated reconstructed image can be represented as $\mathbf{y}(j) = \mathbf{D} \cdot \alpha_0(j)$. The reconstructed error was approximated using $e(j) = ||\mathbf{y} - \mathbf{y}(j)||$ [126]. For all classes which are $j \leq i$ the reconstruction error was large while the reconstruction error from the correct class was the lowest. Recognition was achieved by finding the lowest reconstruction error:

$$\text{identity}(\mathbf{y}) = \text{argmin}\{e_i\}. \tag{3.67}$$

3.6 CONCLUDING REMARKS

This chapter presented a thorough technical analysis of different methods that were used to compare against the proposed weighing and segmentation technique. This chapter provided the implementation of the following algorithms: discrete cosine transform, independent component analysis, non-negative matrix factorisation and sparse representation-based classification. It was of paramount importance to implement these methods so as to make the comparison between these methods and the proposed technique have any significant meaning. This meant that the all conditions including pre-processing techniques, computer efficiency, hardware, databases, illumination variations, occlusions and poses were exactly the same. The results obtained from the implementation of these algorithms are presented in Chapter 5 along with the results obtained from the proposed weighing and segmentation technique. This makes the presentation seamless.

CHAPTER 4 PROPOSED WEIGHING AND SEGMENTATION TECHNIQUE

4.1 CHAPTER OVERVIEW

This chapter presents the complete methodology used to obtain the results of the proposed weighing and segmentation technique for this dissertation. In Section 4.2 the theoretical and technical side of the algorithm is explained so as to fully understand its workings. It focuses on the techniques used to improve a recognition system. The segmentation technique is also presented and the models used to train the algorithms are also explained and presented. These models are subsequently used in experiments to gauge the performance of the technique. The new weighing technique is presented and explained as well. The segmentation and weighing approaches are combined to make a hybrid technique that improves recognition metrics. In Section 4.3 the three databases that are used to obtain results are presented. These databases include AR database, ORL database and Yale Faces database. These databases contain images with differences in pose, illumination, emotion and partial occlusions.

4.2 WEIGHED AND SEGMENTED LINEAR DISCRIMINANT ANALYSIS

Linear Discriminant Analysis (LDA) utilises Fisherfaces algorithm to extract features, train the model and finally classify the query image. Fisherfaces algorithm is a pattern classification approach which minimises the effects of varying light intensity, poses and different facial expressions [51]. The recognition model worked with only greyscale images. Using greyscale images as compared to using RGB images has the advantage of speeding up processing time 3-fold. This is because RGB images are described by 3 intensity values for any pixel. On the other hand, a greyscale image is only described by 1 intensity value per pixel. This can be seen as compressing the data contained in RGB images. This information loss is insignificant to the detection and recognition phase as they can easily recognise a face in greyscale format. An image is defined as:

$$\mathbf{x}(n_1, n_2). \tag{4.1}$$

where n_1 and n_2 are the amount of pixels in the x and y directions. Since each image is an n_1 by n_2 matrix it can be expressed as a flattened vector containing one row of pixels. The vector's length is given by:

$$N = n_1 \text{ x } n_2. \tag{4.2}$$

4.2.1 Image segmentation

LDA is widely used as a feature extraction technique in facial recognition. There have been several attempts to speed up and increase the accuracy of LDA. Some research has gone into improving the accuracy metrics. Traditional LDA extracts features holistically and treats an image as a singular whole feature vector. There have been attempts to divide the face into modules and consider the influence of different regions of the face [135]. The two-dimensional LDA (2DLDA) does not decompose the image into a column vector, it treats the image as a 2D matrix of features. This 2D matrix is used to construct the covariance matrix for feature extraction without the need of decomposing the original image into one dimension. This led to an increase in computational efficiency in extracting features [136]. Finally modular 2D LDA takes advantage of both the 2D approach and the modular approach as it fuses them together to increase computational efficiency and local feature extraction.

Figure 4.1. Image subdivision based on MPCA [135].

Although these methods intended to increase the efficiency they still suffer from using too many regions of the face. They divide the face into several regions, in an unsystematic manner, without actually taking into account the regions of the face that offer the most discrimination. These methods also use all the regions of the face, even those that play little to no role in face recognition. Figure 4.1 shows how these papers achieve this modular approach. As in the figure, some parts of the face like the hair, chin, parts of the cheek and the upper forehead offer little discrimination. Since these parts are generally monotone they are affected by illumination conditions and since there no variations in the colour they do not contribute to much discrimination.

So there is a dire need to develop a technique that takes these shortcomings into account. This dissertation introduces a technique that segments the face in a smarter way. It discards those regions of the face that offer little discrimination and and keep only the regions that do offer the most discrimination.

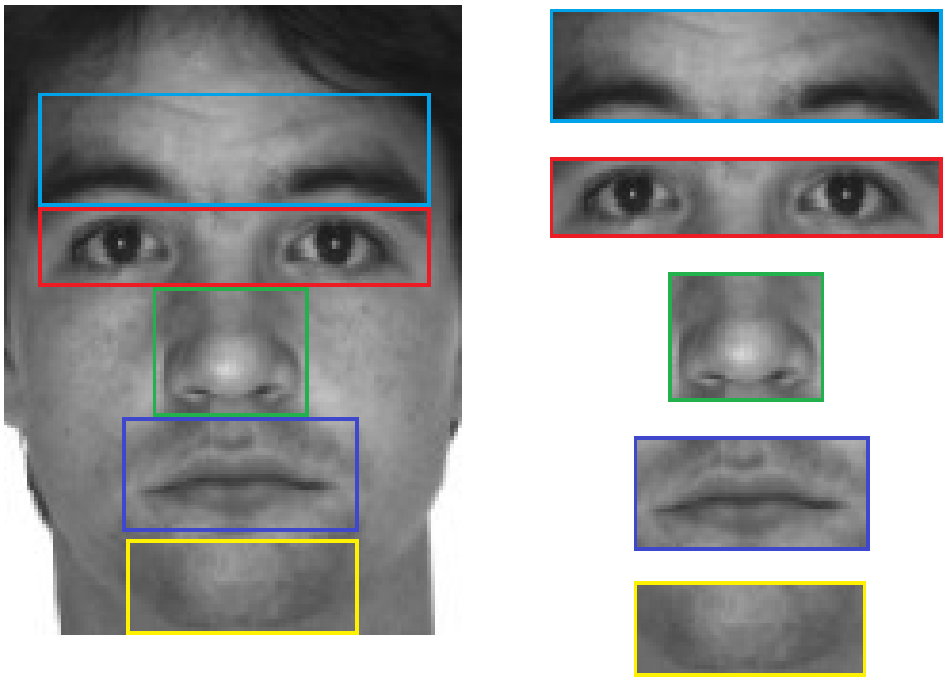


Figure 4.2. Proposed image segmentation scheme.

Figure 4.2 shows how an image was divided into five blocks that were consequently used for face recognition. These individual blocks were also used as building blocks for five individually trained models for recognition. This meant that there was a model for the eyes, a model for the nose, a model for the mouth, a model for the forehead and a model for the chin. Different permutations of these models were combined to make different combinations. These permutations were also used for recognition.

In order for the technique to obtain the coordinates of the key features, cascades were used to identify the three main features which are the eyes, the nose and the mouth. After these regions were detected the forehead region and chin region were pragmatically obtained by using the average values of each class in the training set.

In total, there were 31 models that were constructed using different combinations of the five selected features of the face. These models were used in conjunction with Fisher's discriminant in order to recognise a face. One thing has to be highlighted, 31 models were built and these were exhaustive considering the five features used.

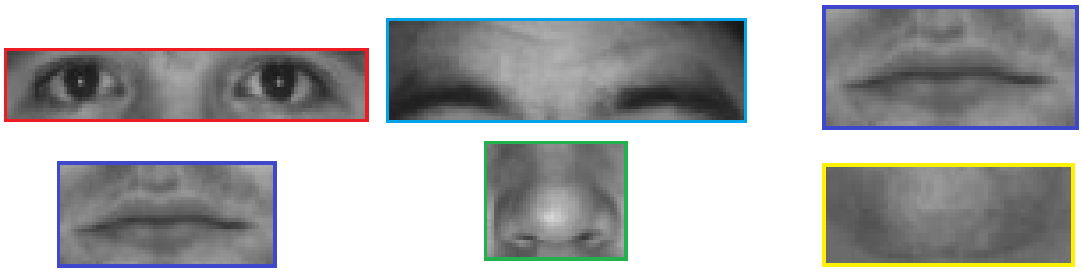
Table 4.1. Legend.

Letter	Model
C	Chin
E	Eyes
F	Forehead
M	Mouth
N	Nose

Table 4.2. Combinations of different models.

	Category 1	Category 2	Category 3	Category 4	Category 5	
	C	CE	CEF	CEFM	CEFMN	
	E	CF	CEM	CEFN		
	F	CM	CEN	CEMN		
	M	CN	CFM	CFMN		
	N	EF	CFN	EFMN		
		EM	CMN			
		EN	EFM			
		FM	EFN			
		FN	EMN			
		MN	FMN			
Total	5	10	10	5	1	31

Figure 4.3 shows some of the possible combinations that were used. The following sub sections detail the technical side and how recognition was obtained.



(a) (b) (c)

Figure 4.3. (a) The eye and mouth model. (b) The forehead and nose model. (c) The mouth and chin model.

Figure 4.4. Concatenated features that represent a Category 4 model: ENMF.

Figure 4.4 shows the result of concatenating 4 features, the forehead, eyes, nose and mouth, into a Category 4 model called **ENMF** model. This figure looked illegible because the pixel values from each individual feature were flattened first and then concatenated into a one-dimensional vector of length 12086. The vector was then reshaped into a two-dimensional array that could be easily displayed. Each individual subject forms a different pattern when all the features have been flattened and reshaped into a two-dimensional vector. This vector is uniquely different from other images in both the training set and testing sets. These different vectors then formed both the testing and the training set.

4.2.1.1 Image Conversion

Testing and training images were converted from a two-dimensional matrix representation into a one-dimensional vector representation. The first row of the image's pixels was converted into a one-dimensional row vector, the second row was concatenated into the vector and so forth until all n_1 rows had been concatenated into the vector. After the vector had been constructed it was transposed to make a 1-dimensional column vector.

A set of M training images $\{x_1, x_2, ..., x_M\}$ was obtained, each image had a resolution of n_1 by n_2. Each pixel is considered as a single feature in the N dimensional vector space [51], [28]. Thus the k-th image in the training set of image is represented by [28]:

$$x_k = (x_{k,1}, x_{k,2}, x_{k,3}, ..., x_{k,N})^T. \tag{4.3}$$

For LDA to work, it is required that there is at least 1 picture for any given individual. All the images in the training set belong to any of the classes $\{C_1, C_2, C_3, ..., C_g\}$. Each class in the training set represents 1 unique individual. Therefore the total number of images in the training set is given by:

$$M = M_1 + M_2 + M_3 +, ..., M_g. \tag{4.4}$$

4.2.1.2 Calculating the mean and scatter

The next step is to calculate the mean of a single class. Fisherfaces ulitises LDA which main aim is to reduce the dimensionality of the training class to a lower dimension while at the same time conserving all the discriminatory information that best separates different classes from each other [51]. LDA selects some features or discriminants that do the best job of separating the facial images into their different classes in the feature space. This feature space is the low dimensional space.

The aim is to minimise the variations within a specific class whilst maximising the variations between classes. This means that when images belonging to a single class are projected to the discriminant space they cluster together with minimum separation, while the separation between the classes is maximised. The mean of a single class is defined as:

$$\mu_i = \frac{1}{M_i} \sum_{i=1}^{M_i} x_i, \tag{4.5}$$

where, x_i defines a single image, M_i is the total number of images in class i and μ_i is the mean of class i. When all the images in class C_i have been converted into a 1 dimensional vector, the mean of all the pixels at location 1 for the images in that class is found, the procedure is repeated until the means of all the pixels up to location N. Equation (4.6) shows how this procedure is done.

$$\mu_i = \frac{1}{M_1} \begin{pmatrix} x_1^1 & +x_1^2 & +\cdots+x_1^{M_1} \\ x_2 & +x_2 & +\cdots+x_2^{M_1} \\ \vdots & \vdots & \vdots \\ x_N^1 & +x_N^2 & +\cdots+x_N \end{pmatrix}. \tag{4.6}$$

The first column corresponds to the first image in class C_i. M_1 is the number of images in class C_i. N is the number of pixels in a single image

The next step is to subtract the class mean μ_i from all the images in that class. This is repeated for every class in the training set $\{ C_1, C_2, C_3, ..., C_g \}$. Equation (4.7) shows how this is done [51]:

$$\Phi_i = x_k - \mu_i. \tag{4.7}$$

Φ_i is vector i which represents image i that as the classes mean μ_i subtracted from it. This effectively means the most common features within that specific class have been removed from that particular class. It becomes much easier to discriminate between different images.

(a) Mean image of class 2. (b) Mean image of class 3

(c) Mean image of class 13. (d) Mean image of class 17

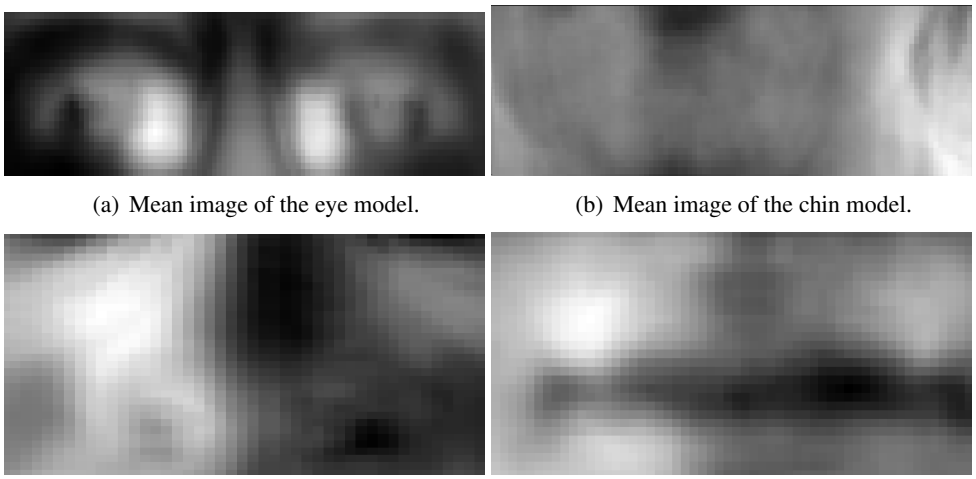
Figure 4.5. The mean images from the four different classes before 31 models were extracted.

Figure 4.5 shows the results of subtracting the mean image of a class from all images in a particular class. Figure 4.5 (a) is an example image before the mean was subtracted. Figs. 4.5 (b), (c) and (d) shows 3 images from a single class after the mean was subtracted from the images.

The mean of all the classes is then calculated according to:

$$\mu = \frac{1}{g}\sum_{i=1}^{g}\mu_i, \tag{4.8}$$

where g is the total number of classes in the training set. This μ is the mean image of the entire training set. Figure 4.6 shows the mean images of four different models found when the detection model was trained using the AR Database.



(a) Mean image of the eye model. (b) Mean image of the chin model.

(c) Mean image of the nose model. (d) mean image of the mouth model.

Figure 4.6. Mean images of the different models.

The next step is to find the scatter matrix of all the classes. Between class mean S_B is defined as [51]:

$$S_B = \sum_{i=1}^{g} M_i \, (\mu_i \text{ - } \mu)(\mu_i \text{ - } \mu)^T. \tag{4.9}$$

In (4.9), M_i represents the number of samples in the corresponding class i, and g is the total number of classes in the training set. The next step is to find the within class variation S_W. To find this scatter matrix, (4.10) is used [51]:

$$S_W = \sum_{i=1}^{g} \sum_{x_k \in C_i} M_i \, (\mu_i \text{ - } \mu)(\mu_i \text{ - } \mu)^T. \tag{4.10}$$

Fisher's Linear Discriminant (FLD), proposed by Ronald Fisher, is a class specific method in that it has a-priori knowledge of which images belong to which classes and thereby shapes the resulting scatter in a particular way that makes it possible to separate all the given classes linearly [51]. FLD reduces the image space from $(n_1 \cdot n_2) \cdot g$ where g is the number of total classes in the training set and $n_1 \cdot n_2$ is the number of pixels, to just g-1 dimensions. This is a significant reduction in the total dimensionality of the data [51].

FLD manages to separate classes linearly selecting an optimal projection W_{opt} that maximises Fisher's optimisation criteria. The aim is to maximise the ratio of between class scatter, S_B, and within class scatter, S_W [51]. This is done by [51]:

$$W_{opt} = [w_{opt,1}, w_{opt,2}, w_{opt,3} \ldots, w_{opt,m}] = arg\max \left\{ \frac{W^T S_B W}{W^T S_W W} \right\}, \tag{4.11}$$

where m corresponds to the largest eigenvalues that produce the largest eigenvectors.

The next step is to find the largest eigenvectors that describe the discriminant space. Eigenvectors are also knows as an eigenface.

$$arg\max\left\{\frac{W^T S_B W}{W^T S_W W}\right\} \Leftrightarrow (S_W^{-1} S_B \text{ - } \lambda_i) w_{opt,i} = 0. \tag{4.12}$$

W is a square matrix, $S_W^{-1} S_B$ is symmetric matrix which has orthogonal eigenvectors. The eigenvalues are used to find eigenvectors by solving [51]:

$$(S_W^{-1} S_B \text{ - } \lambda_i) w_{opt,i} = 0, \tag{4.13}$$

which reduces to the following eigen problem:

$$S_B w_{opt,i} = S_W^{-1} \lambda_i w_{opt,i} \quad | \quad i = 1,2,3,...,m. \tag{4.14}$$

4.2.1.3 Sorting eigenvectors

This process produces m eigenvectors, m is the number of images in the training set. Most of the energy is contained in the first 10-20 Eigenvectors and the first eigenvector contains the most energy. All the eigenvectors are sorted in descending order and the first m eigenvectors were selected, in this case 15 eigenfaces were selected. After 15 eigenvectors, the produced eigenfaces no longer resembled a face as the remaining eigenvectors capture most of the noise. Fifteen eigenfaces could approximate and reconstruct the face.

All images in the training set can be represented by a linear combination of a finite number of eigenvectors, m. Equation (4.15) sums up how the faces are projected int the eigenspace [51]:

$$\Phi_i = \sum_{j=1}^{m} w_j \cdot W, \tag{4.15}$$

where, w_j is equal to $W_j^T \cdot (x_k - \mu)$, u_j is the eigenvector corresponding to index j, and Φ_i is the the average face. Figure 4.7 - Figure 4.11 shows the reconstructed eigenfaces. These eigenvectors represent 5 of the 31 models that were constructed for face recognition, they are a subset of the whole. These eigenfaces are the features that make up up any query image. These can be thought of as the features that are extracted by LDA after projection.

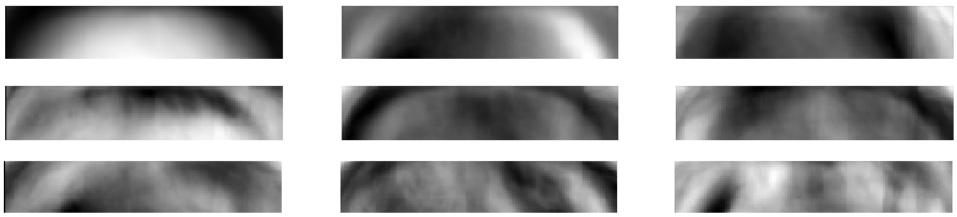


Figure 4.7. The forehead model's first 9 eigenvectors.

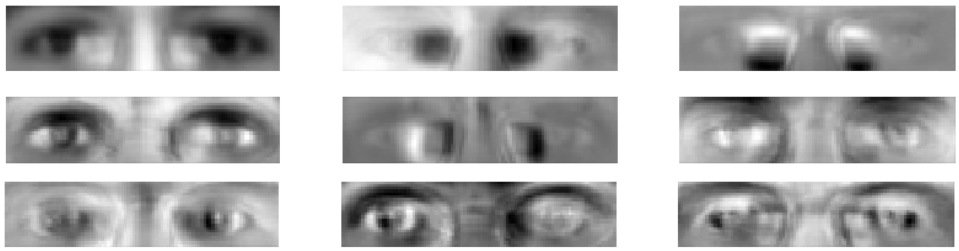


Figure 4.8. The eye model's first 9 eigenvectors.

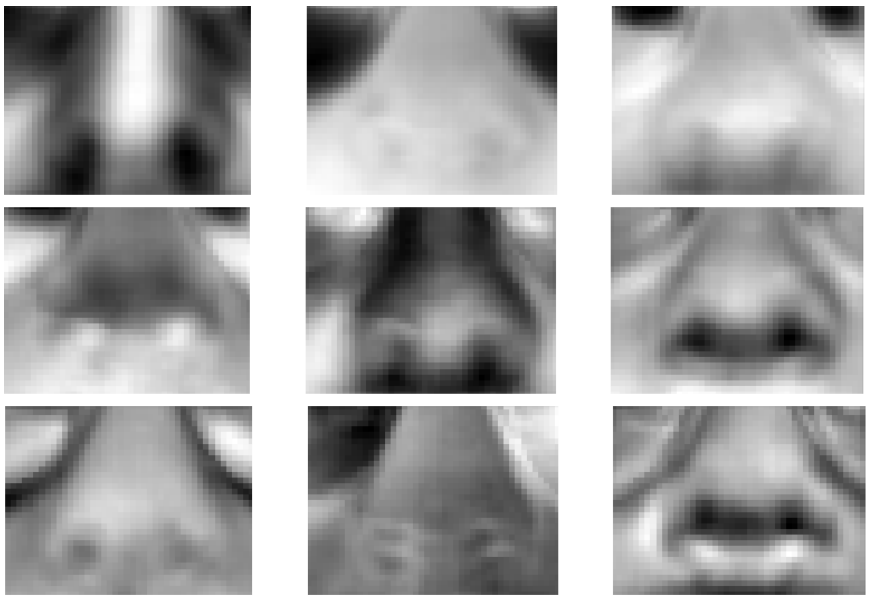


Figure 4.9. The nose model's first 9 eigenvectors.

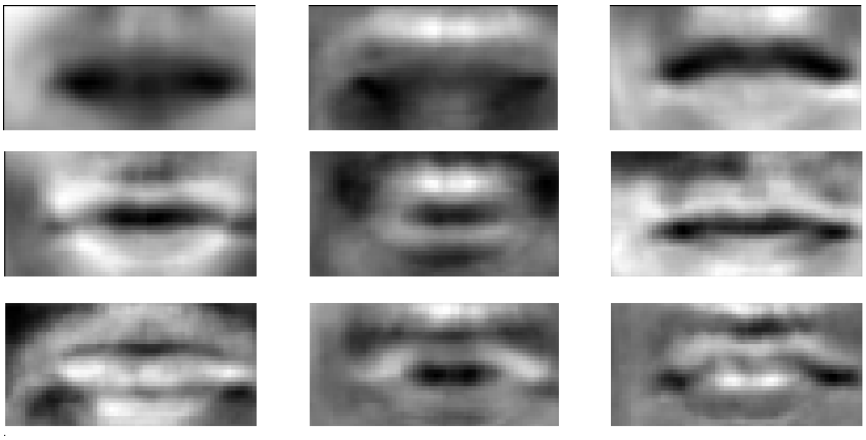


Figure 4.10. The mouth model's first 9 eigenvectors.

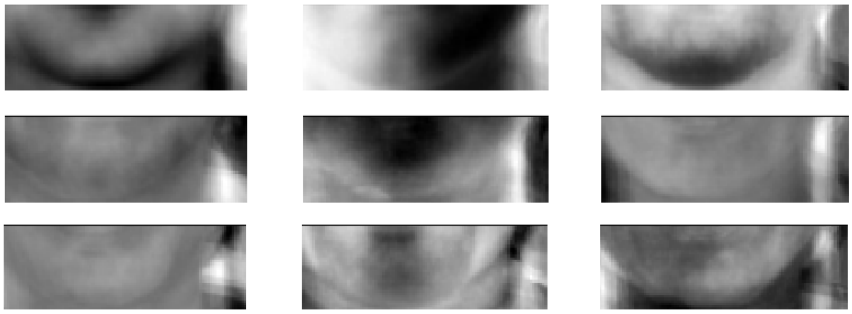


Figure 4.11. The chin model's first 9 eigenvectors.

Since an image can be decomposed and represented by only eigenvectors, each single eigenvector is part of the group of eigenvectors that can be used to represent a face fully. Each and every projection w, in (4.16), is a vector that is a result of calculation using an eigenvector and an image. To represent a face, K = 15 eigenvectors were used thus a face could be mathematically represented as [51]:

$$\Omega_i = \begin{bmatrix} {w_1}^i \\ {w_2}^i \\ \vdots \\ {w_k}^i \end{bmatrix} \quad i = 1,2,3,\ldots,M, \tag{4.16}$$

where, w_1^i is the image that corresponds to the eigenvector located at index 1, K is the number of eigenvectors used, and Φ_i is the average face. The last step in building the recognition model is to identify a query image. The query image, x_{query}, is converted into a 2-dimensional matrix of integers and then into a 1-dimensional column vector given by:

$$x_{\text{query}} = (x_{\text{query},1},\ x_{\text{query},2},\ x_{\text{query},3},\ldots,\ x_{\text{query},N})^T, \tag{4.17}$$

where, $x_{\text{query},1}$ is the first pixel in he column vector and $x_{query,N}$ is the last pixel value in the column vector. The next stage is to subtract the mean image from the query image:

$$\Phi = x_{\text{query}} - \mu. \tag{4.18}$$

The image is then projected into eigenspace to give its discriminant vector, Ω using the following formula [51]:

$$\Omega = W^T \cdot \Phi, \tag{4.19}$$

Ω then becomes:

$$\Omega = \begin{bmatrix} {w_1}^i \\ {w_2}^i \\ \vdots \\ {w_k}^i. \end{bmatrix} \tag{4.20}$$

4.2.2 Recognition

The next step is to is to find the class closest to the query image, x_{query}, this is achieved by using the Nearest Neighbour discriminant rule:

$$min \parallel \ \Omega \text{ - } \Omega_i \ \parallel = \varepsilon_j < \theta_j, \tag{4.21}$$

where, ε_j is the distance of query image from class C_j, θ_j is the maximum threshold distance any query image must be in, in order to be assigned to class C_j, and i ranges from 1 to the number of classes in the training set. $i \in [1, 2, 3, ..., g]$.

When recognising a query image there are two scenarios that arise from using the Nearest Neighbour discriminant rule:

1. $\varepsilon_j < \theta_j$: The query image belongs to class C_j. Therefore the face belongs to a registered user.
2. $\varepsilon_j > \theta_j$: The query image does not belong to any class therefore it belongs to an unregistered user.

4.2.3 Weighing technique

LDA deals with projecting a high dimensional image into a lower dimensional feature space. During classification all features in the lower dimensional space are treated as having the same effect on classification, of which that is not the case. Some features are critical in classifying any given image,

thus there is a need to have some sort of function that can highlight the most important features (e.g, nose, mouth, eyes, chin and ears). These key features are used primarily for identifying any face.

The existing literature only deals with a single centre which is located in between the two eyes. This has a draw back of only highlighting the features near the eyes as the most important. This dissertation proposes that not only are the eyes the only key features during face recognition, but the nose, the mouth, part of the chin and part of the forehead region also play a crucial part in face recognition. This dissertation thus makes use of five centres namely: the centre of the two eyes, the middle of the brows, the nose and the mouth.

There are various functions that can be used as a means to weight each individual facial feature. Some of these functions include the Gaussian function, radial basis function (RBF), radial basis function network (RBFN), polynomial kernel, Obst kernel network. The Gaussian function and the radial basis functions were used as the weighing functions in this dissertation. When using WLDA alone the calculation of the between class scatter would be multiplied by a weighing function. The calculation thus becomes:

$$S_B = \sum_{i=1}^{g-1} \sum_{j=i+1}^{g} M_i \, M_j \omega \, (\mu_i \text{ - } \mu_j)(\mu_i \text{ - } \mu_j)^T . \tag{4.22}$$

where, M_i and M_j are class priors.

4.2.3.1 Gaussian function

A Gaussian function is a mathematical function frequently used to represent a random variable's probability density function. This random variable has a normal distribution. The Gaussian function graph is symmetric about its vertical axis and resembles a bell curve. These functions are frequently used to describe normal distributions in statistics. They are also useful in image processing when they are utilised as Gaussian blurs. Diffusion and heat equations also makes use of Gaussian functions in order to find solutions. The function can be defined as:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp\left[-0.5\left(\frac{x-\mu}{\sigma}\right)^2 \right], \tag{4.23}$$

where, x is an arbitrary random value, μ is the expected value, σ is the variance, and π is a mathematical constant.

Gaussian functions are often used in one dimensional data. They can be modified to cater for higher dimensions. In this dissertation, they were modified to be a two-dimensional function that caters for images. Where x and y are the horizontal and vertical coordinates of any arbitrary pixel. Thus the Gaussian function used became:

$$\omega(i,j) = \exp\left[-\frac{\left((i-x)^2+(j-y)^2\right)}{\sigma^2}\right]. \tag{4.24}$$

The eclosion centre refers to the two centres of the eyes, nose, mouth and centre of the brows. On that basis, this dissertation established a modified version that utilised the Gaussian function and had five centres on both the centres of the eyes, the centre of the brows, the nose and mouth:

$$\begin{aligned}\omega(x,y) = \exp&\left[-\frac{\left((x-i_1)^2+(y-j_1)^2\right)}{\sigma_1^5}\right]^{\frac{1}{5}} \\ &\cdot \exp\left[-\frac{\left((x-i_2)^2+(y-j_2)^2\right)}{\sigma_2^5}\right]^{\frac{1}{5}} \\ &\cdot \exp\left[-\frac{\left((x-i_3)^2+(y-j_3)^2\right)}{\sigma_3^5}\right]^{\frac{1}{5}} \\ &\cdot \exp\left[-\frac{\left((x-i_4)^2+(y-j_4)^2\right)}{\sigma_4^5}\right]^{\frac{1}{5}} \\ &\cdot \exp\left[-\frac{\left((x-i_5)^2+(y-j_5)^2\right)}{\sigma_5^5}\right]^{\frac{1}{5}},\end{aligned} \tag{4.25}$$

where, (x,y) is the position of a pixel and (i,j) is it's eclosion centre. This Gauss function aids in enhancing critical classification information on the facial region, thus enhancing the segments of the face where discriminating features are located. Increasing the value variance, σ^2, increased the spread of the Gaussian distribution, this meant that the weights of features further away from the centres approached 0. Thus they had little to no effect on discrimination. By tuning the variance of the five selected features σ_1^2, σ_2^2, σ_3^2, σ_4^2, σ_5^2, this resulted in increasing or decreasing the weights of the five selected features. In this dissertation the variance of the different features was as follows:

Table 4.3. A description of which variance corresponds to which feature.

Variance	Feature
σ_1^2	left eye
σ_2^2	right eye
σ_3^2	centre of brows
σ_4^2	nose
σ_5^2	mouth

4.3 DESCRIPTION OF DATABASES USED

A. AR Database

The AR Face database was used for both training and testing purposes [87]. This database consists of over 3000 images of 1226 subjects half of them are men and the other half are women. For each subject, there were 26 images for that specific class. The images varied with light intensity, pose, expression and occlusions. The occlusions included scarfs and glasses. All images were converted to grayscale.

Figure 4.12. Sample images from the AR database [87]

Figure 4.12 shows some of the images used for testing and training images for the experiments. For all experimental results 15 male subjects and 15 female subjects were used totalling 780 images. The database was split into a ratio of 0.7:0.3 between the training set and testing set resulting in 624 training images and 156 testing images.

B. ORL Database

The ORL database, (also known as the AT & T database of faces), consists of a total of 400 images [88]. Within these images there are 40 distinct subjects where each subject was taken 10 images. The captured images consist of subjects who were smiling, not smiling, had glasses on, had glasses off, different illumination conditions and were taken against a dark homogeneous background. Figure 4.13 shows samples taken from the ORL database.

Figure 4.13. Sample images from the ORL database [88].

C. Yale Faces Database

The Yale faces database was used as a bench marking database as it did not contain any occluded images [89]. It contains grayscale images of 15 subjects having 11 images each. It has a total of 165 images, these images vary with light intensity, emotion and poses. All the images were captured on the same white background. The same split ration of 0.7 was used for splitting the data into training samples and testing samples. In this case 132 images were used for training purposes while 33 were used for testing purposes.

Figure 4.14. Sample images from the Yale faces database [89].

Database	Total number of images	Unique individuals	Images per individual	Occlusions
AR [87]	520	20	26	Yes
ORL [88]	400	10	15	No
Yale faces [89]	165	15	10	No
Total	1085	45		

4.4 CONCLUDING REMARKS

This chapter discussed segmenting an image into 5 features namely; the eyes, nose, mouth, chin and forehead. The rest of the features of a face like the cheeks, ears and the hair were discarded owing to the fact that not much discrimination can be obtained from those parts of the face. 31 models, which comprised of a combination of these five features, were built and LDA was used to train these models. Euclidean distance measure was utilised for classification. Several papers focus on the face as a whole when extracting features, they do not focus on individual regions to see how they affect face recognition [26], [30], [53], [60], [62], [112], [113], [137], [138], [139]. This technique of segmenting the face into different sub-regions and studying how these combinations affect recognition is lacking in pre-existing literature. This chapter aimed to fill in that gap in knowledge hence the use of 31 models to comprehensively study that problem and put that question to bed.

Not all regions of the face contribute equally to discrimination. It has been theorised that the eyes and nose play the leading role when it comes to discrimination. This chapter also discussed a weighing technique that assigned weights to different regions of the face according to features. Other papers only studied the centre of the eyes as the focal point where weights should be assigned. This chapter put forward a new technique in which 5 regions of the face were identified as key focal points. These focal points were called ecclosion centres. These focal points were then used as the centres of where the weighing technique started assigning weights to the rest of the face. Once the rest of the face was assigned weights the different class specific models were trained for each subject. Once training had finished the recognition process was done in order to get the accuracy metrics of this weighing technique. The databases used for comparison tests were also described, these databases include images with varying features and characteristics. From databases that included different illumination conditions, face orientation, pose, occlusions and expression.

CHAPTER 5 RESULTS AND DISCUSSION

5.1 CHAPTER OVERVIEW

This chapter presents the results obtained from the experiments carried out in the previous chapters. These results were obtained from a qualitative and quantitative study on how the various combination of regions on the face affect the recognition rate and efficiency of the segmentation approach. In Section 5.2 the notion that is used to refer to the models that were built is presented. The evaluation metrics that are used to understand the obtained results are also presented. In Section 5.3 the results obtained using the AR database are presented. A discussion of the results is also presented. In Section 5.4 the results obtained using the AR database are presented. A discussion of the results is also presented. In Section 5.5 the results obtained using the AR database are presented. A discussion of the results is also presented. In Section 5.6 the average metrics and results obtained across all databases are presented. These results are compared against the four methods that were implemented in Chapter 3. In Section 5.7 the comparison of results obtained from the proposed technique and those taken from literature is presented. For the proposed technique to be considered average, better or excellent this comparison is necessary.

5.2 UNDERSTANDING THE NOTATION OF THE RESULTS

The proposed image segmentation technique focused on the five prominent regions of that face, namely the chin, eyes, forehead, mouth and nose. These five regions were given the notation:

- C - chin
- E - eyes
- F - forehead
- M - mouth
- N - nose.

Thirty-one models were constructed by using combinations of these five regions. These combinations include the chin and forehead, the forehead, mouth, nose and the eyes, mouth, nose and forehead. Table 4.1 shows all the 31 models that were built. The algorithm was trained on these models and

accuracy metrics were calculated.

The Category simply means how many features were actually used for the segmentation approach. For example, a Category 3 model consists of 3 sub-models.

5.2.1 Evaluation

Evaluation for classification algorithms usually takes several forms which mainly include accuracy and error rates. For this dissertation the following metrics were used:

- **True positive (TP)** or **recognition rate (RR)** is the number of images that were classified as belonging to their true actual class.
- **False negative (FN)** is the number of images that were incorrectly classified as belonging to a different.
- **False positive (FP)** is the number of unregistered images that were classified as belonging to the training database.
- **True negative (TN)** is the number of registered images that were incorrectly classified as not part of the training database.
- **Precision** measures how correct the positive prediction is:

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}}. \tag{5.1}$$

- **Recall** also known as true positive rate (TPR) or Sensitivity. It measures the ration of the positive images that were classified as positive:

$$\text{Recall} = \frac{\text{TP}}{\text{TP + FN}}. \tag{5.2}$$

- **Accuracy** is a measure of the total number of all correctly classified images, which includes TP rate and TN rate, against all classifications that took place:

$$\text{Accuracy} = \frac{\text{TP + TN}}{\text{TP + TN + FP + FN}}. \tag{5.3}$$

- **Specificity** known as true negative rate (TNR). Measures the correctness of the classifier's negative predictions:

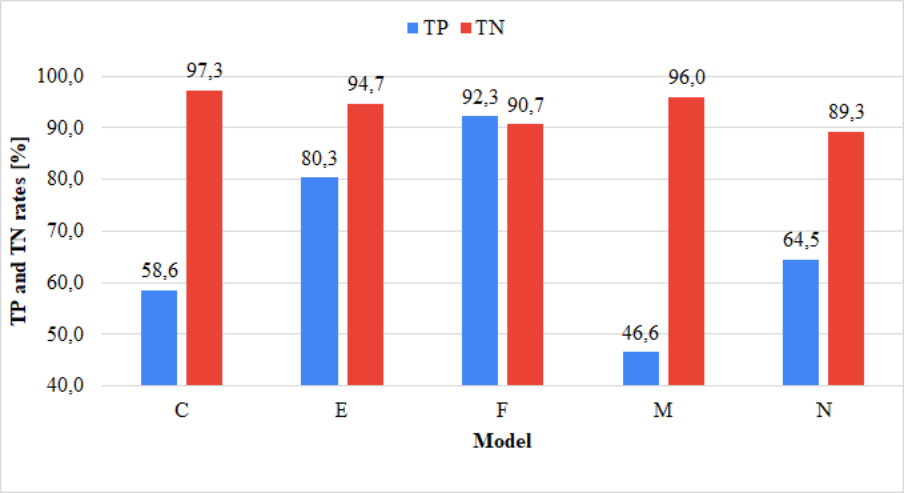
$$\text{Specificity} = \frac{\text{TN}}{\text{TN + FP}}. \tag{5.4}$$

- **F1 score** measures the weighted average of the precision and recall.

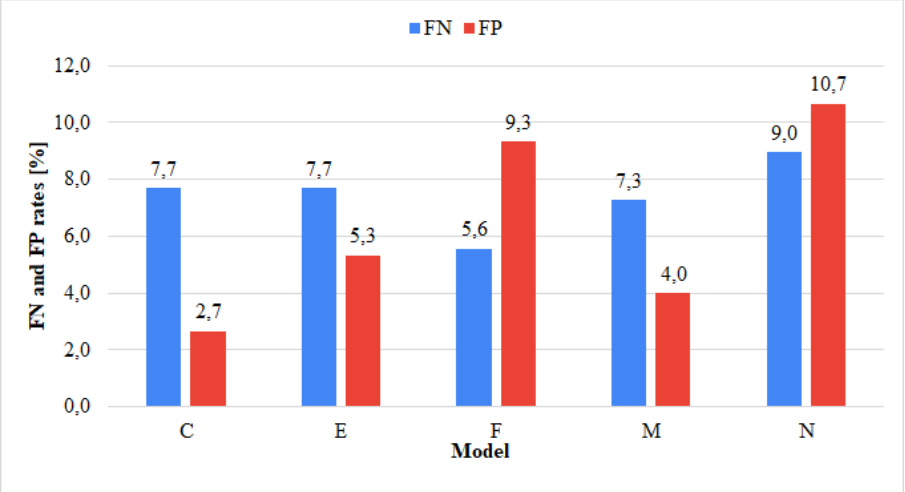
$$\text{F1 score} = 2 \cdot \frac{\text{precision} \cdot \text{ recall}}{\text{precision + recall}}. \tag{5.5}$$

5.3 AR DATABASE PERFORMANCE

5.3.1 Category 1



(a) True Positive and True Negative rates.

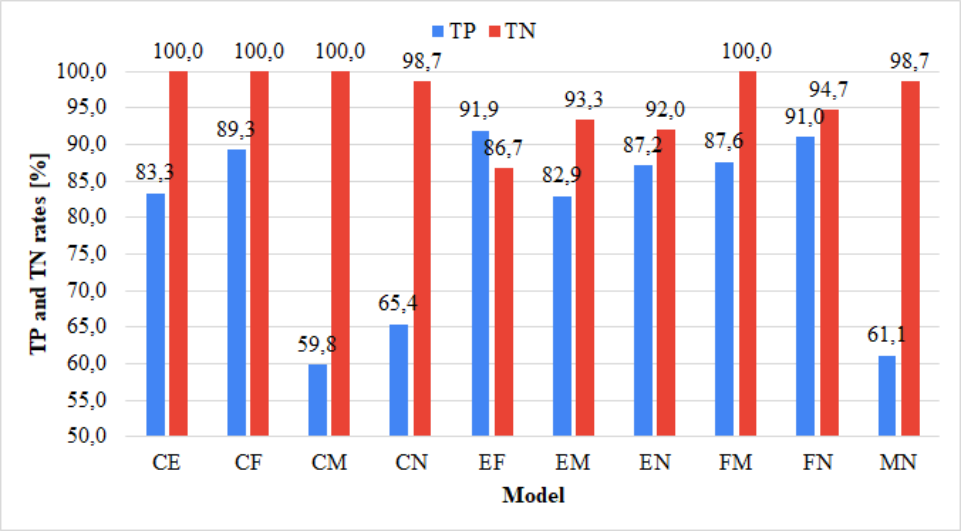


(b) False Positive and False Negative rates.

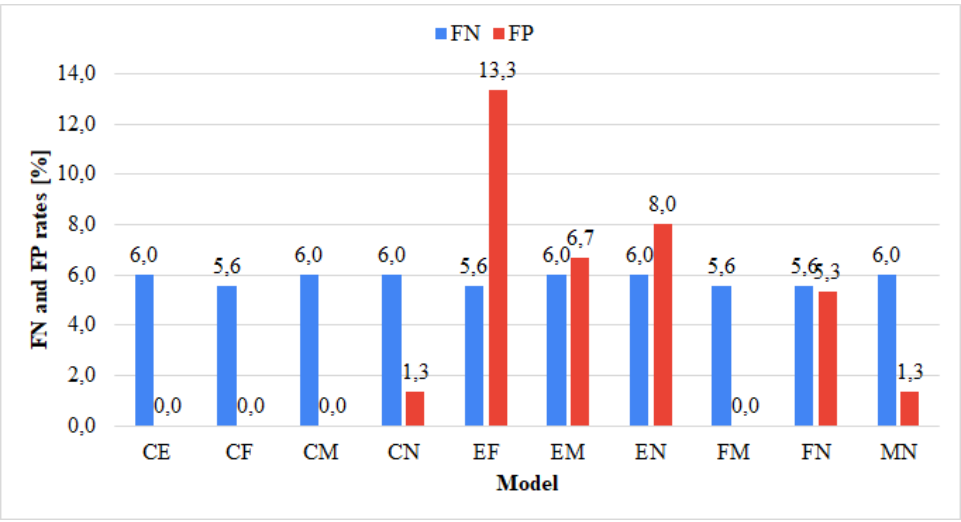
Figure 5.1. The accuracy and error rates for Category 1 models for the AR database.

Figure 5.1 shows the accuracy and error metrics that were obtained on the AR database. The forehead region offered the highest recognition rate with 92.3% while the mouth region had the lowest recognition rate at 46.6%. The average recognition rate of Category 1 models was 68.5%. The true negative rate was relatively high with the lowest being 89.3% and the highest being 97.3%. The false positive rate was low, ranging from 5.6% for the nose model, up to 9% for the nose model. The nose model's false positive and false negative is alarmingly high. The forehead model had the highest recognition rate as well as the lowest false positive rate.

5.3.2 Category 2



(a) True Positive and True Negative rates.

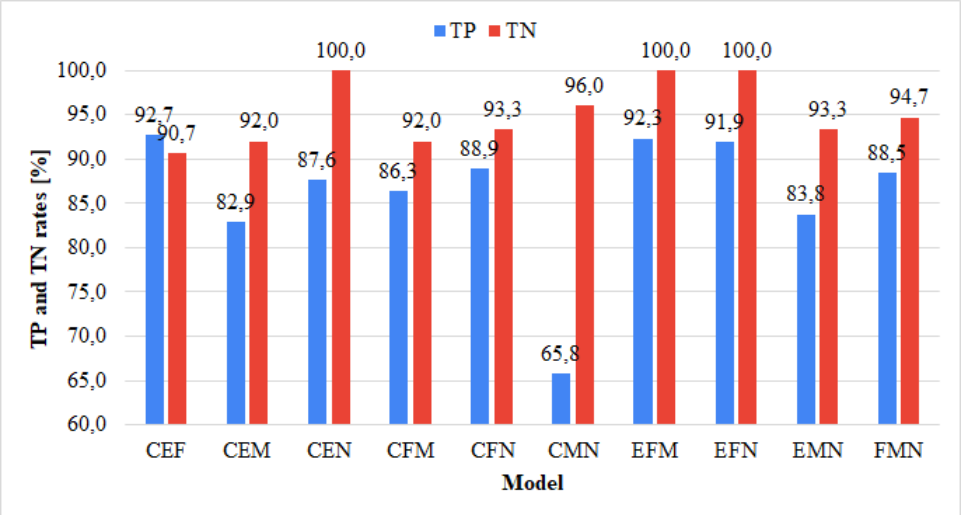


(b) False Positive and False Negative rates.

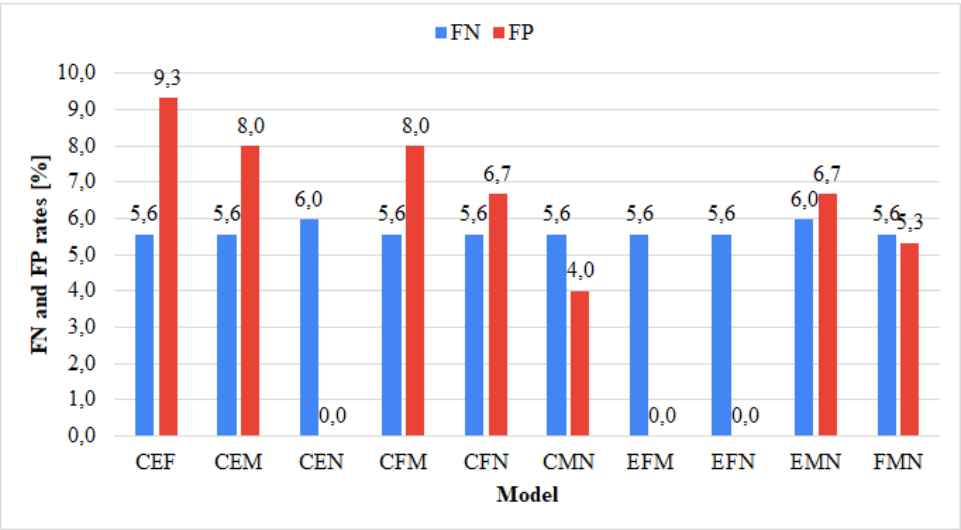
Figure 5.2. The accuracy and error rates for Category 2 models for the AR database.

Figure 5.2 (a) shows the true positive and true negative rate, while Figure 5.2 (b) shows the false positive and false negative rate of Category 2 models. The EF model and the FN model had the highest true positive rate with 91.9% and 91% respectively. The CM models had the lowest true positive rate with 59.8% followed by the MN model. The average true positive rate for Category 2 models was 80%. The EF model had the lowest true negative rate at 86.7% while the CE, Cf, CM and FM models had perfect true negative rates at 100%. The average false negative rate was 5.8% with the CE, CF, CM and FM models with the lowest at 0%. The EF had the highest true negative rate at 13.3%.

5.3.3 Category 3



(a) True Positive and True Negative rates.

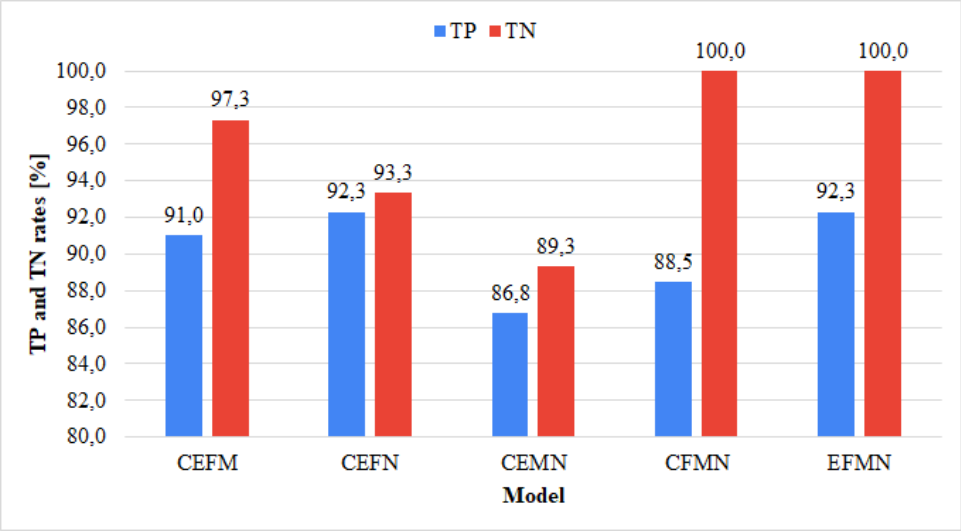


(b) False Positive and False Negative rates.

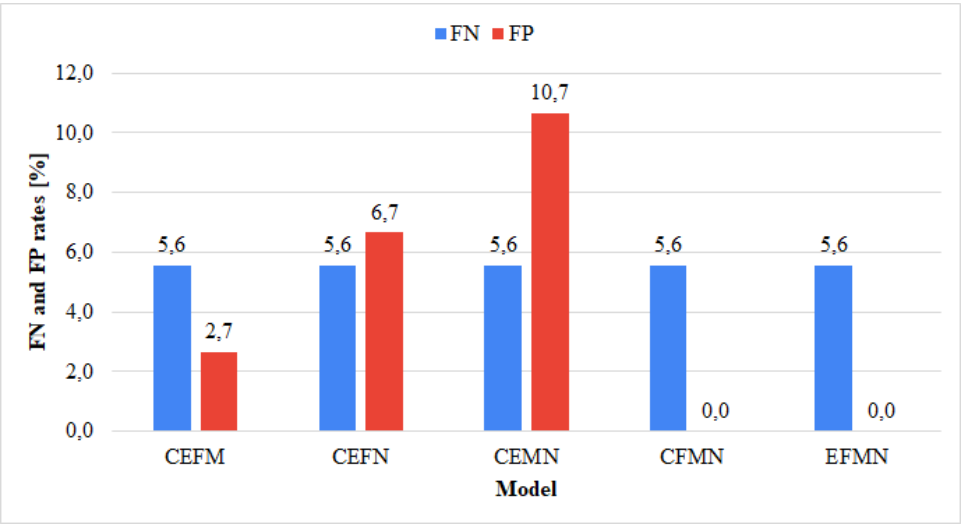
Figure 5.3. The accuracy and error rates for Category 3 models for the AR database.

Figure 5.3 (a) shows the true positive and Figure 5.3 (b) true negative rates for the Category 3 models when compared using the AR database. The average recognition rate was 86%, the average true negative rate was at 95.2%. The CEF, EFM and EFN models had the highest recognition rates at 92.7%, 92.3% and 91.9% respectively. The CEN, EFM and EFN had the highest true negative rates all being at 100%. The false negative and false positive rates were 5.6% and 4.8% respectively. The CEN, EFM and EFN had the lowest false positive rates which were all at 0%. Eight-out-of-ten models had a false positive rate of 5.6% while the remaining two models (CEN and EMN) had a rate of 6%.

5.3.4 Category 4



(a) True Positive and True Negative rates.

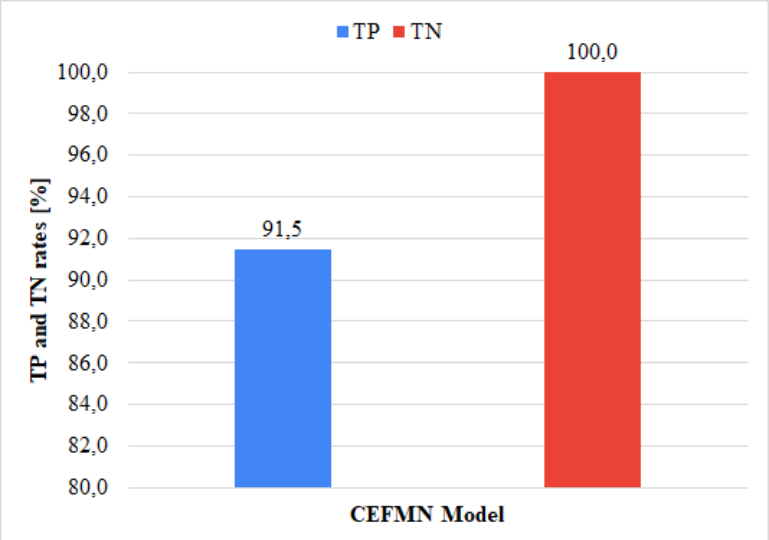


(b) False Positive and False Negative rates.

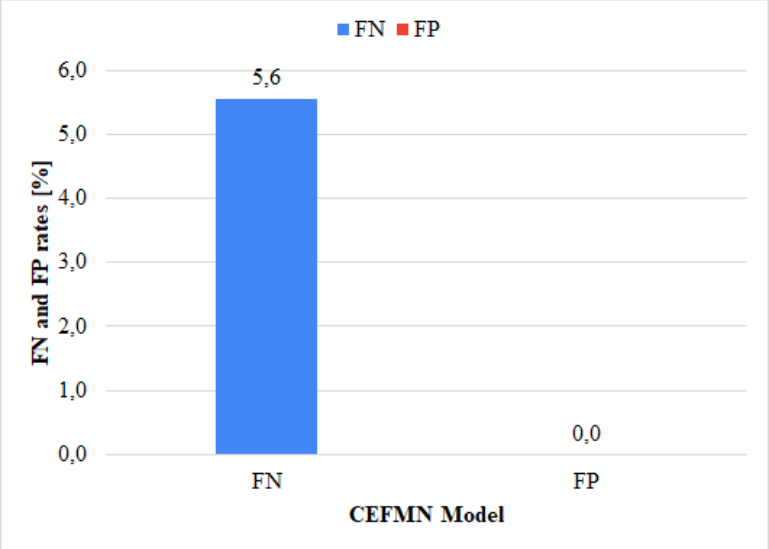
Figure 5.4. The accuracy and error rates for Category 4 models for the AR database.

Figure 5.4 (a) shows the true positive and Figure 5.4 (b) true negative rates for the Category 4 models when comparing using the AR database. The average true positive rate was 90.2% while the average true negative rate was 96%. The CEFN and EFMN models had the highest true positive rate of 92.3% while the CEMN had the lowest at 96.8%. Both the CFMN and EFMN had the highest true negative rate at 100%. The average false negative rate was 5.6% while the average false positive rate was 4%. The CEMN and EFMN had the lowest false positive rate at 0%.

5.3.5 Category 5



(a) True Positive and True Negative rates.

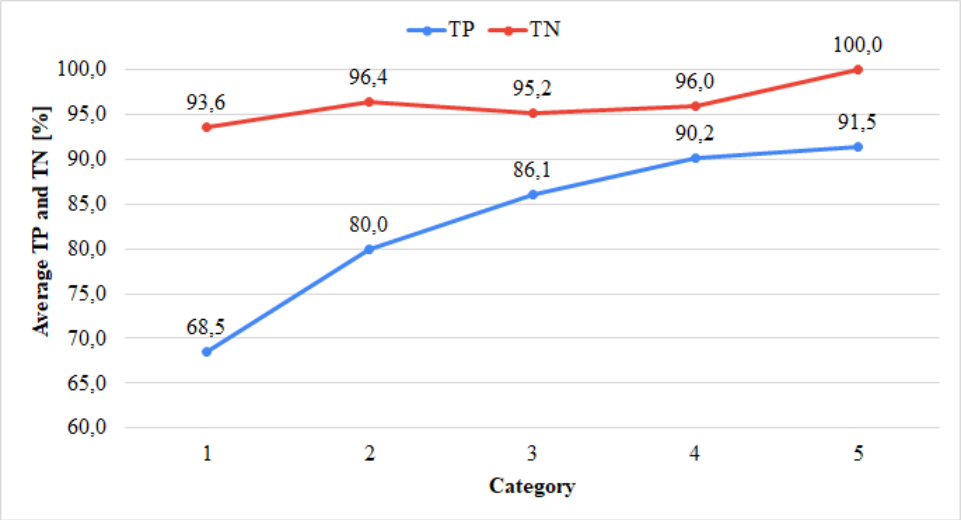


(b) False Positive and False Negative rates.

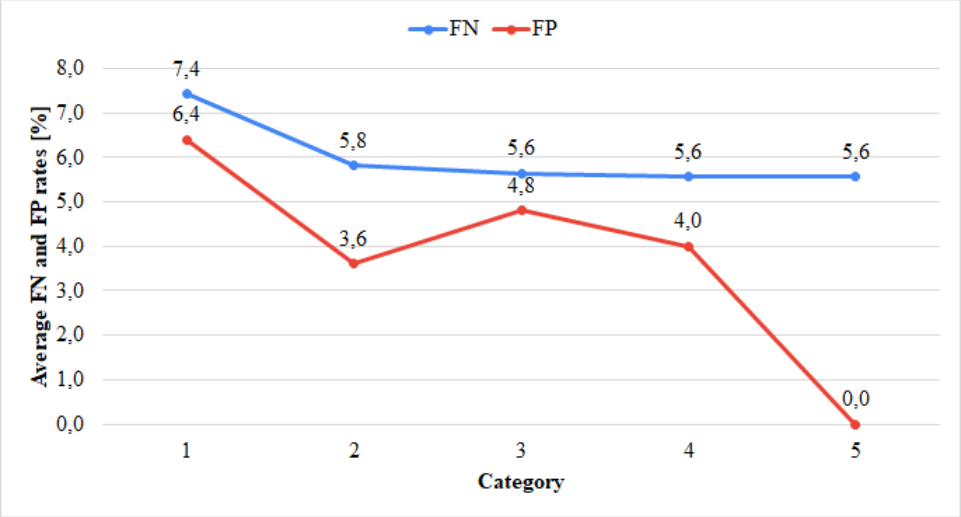
Figure 5.5. The accuracy and error rates for Category 5 models for the AR database.

Figure 5.5 (a) shows the true positive and Figure 5.5 (b) true negative rates for the Category 5 model when comparing using the AR database. The effective recognition rate of the CEFMN model was 91.5% and its true negative rate was 100%. The CEFMN model had a false negative of 5.6% and a false positive rate of 0%.

5.3.6 Discussion of results obtained on the AR database



(a) Average true positive and average true negative rates.



(b) False Positive and False Negative rates.

Figure 5.6. The average metrics across the 5 different categories on the AR database.

When using the Category 1 model the forehead region had the highest recognition rate at 92.3%, followed by the eye model at 80.3% while the mouth had the lowest at 46.6%. This shows that when only using a model that utilises only 1 model, the best decision is to use the forehead as a feature as compared to the eyes or nose. Since the mouth had the worst recognition rate, it is better to not use the mouth for recognition purposes. The average recognition rate was 68.5%.

When it comes to the Category 2 models, two models with the highest recognition rate were the EF and FN models at 91.9% and 91%. Interestingly, the forehead feature was present in both of these highest recognition rates. On the other hand, the CM and MN models had the least recognition rates at 59.8% and 61.1% further showing that when the mouth is a sub-feature in a model the accuracy suffers. The average recognition rate was 80%.

The CEF, EFM and EFN models had the highest recognition rates at 92.7%, 92.3% and 91.9% respectively. The trend of having the forehead as a sub-feature in all the models that performed the best continued. Results from the Category 3 results also highlighted another interesting sub-feature; the eyes (E). The combination of the E and F models had the highest recognition rate for the Category 3 model. This suggests that these two different sub-features may offer the highest discrimination. The average recognition rate was 86.1%.

The average recognition rate on the Category 4 models was 90.2%. It's now glaringly obvious that models which contain the forehead (F) and eyes (F) sub-features out-perform all other models. The CEFN EFMN and CEFM models had the highest recognition rates at 92.3%, 92.3% and 91% respectively. The average true negative rate was 96% which shows that Category 4 models were quite robust in identifying subjects that did not belong to a subject in question.

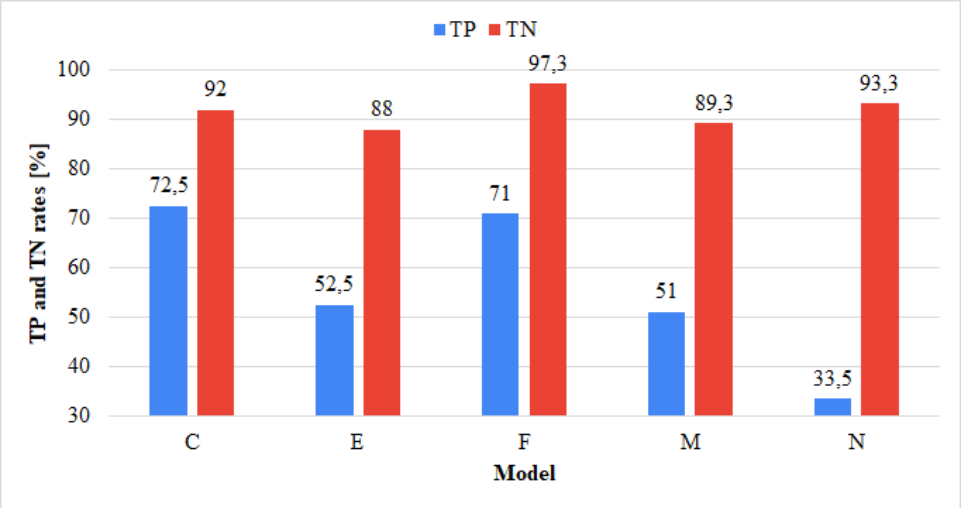
Overly, the Category 5 model which contained all the features had the highest recognition rate as 91.5% and the highest true negative rate at 100% as well. This shows that segmenting the face and applying a weighing algorithm to the sub-regions leads to high recognition rates, but utilising all the sub-features still outperforms Category 1 - 4 models.

Figure 5.6 (a) shows that the more features that are used the better the recognition rate. There only is a 1.3% increase from using Category 4 models to using a Category 5 model. This shows that Category 4 models can be used for recognition purposes when speed and efficiency are desired without sacrificing on recognition. The true negative rate also increases as more features are used.

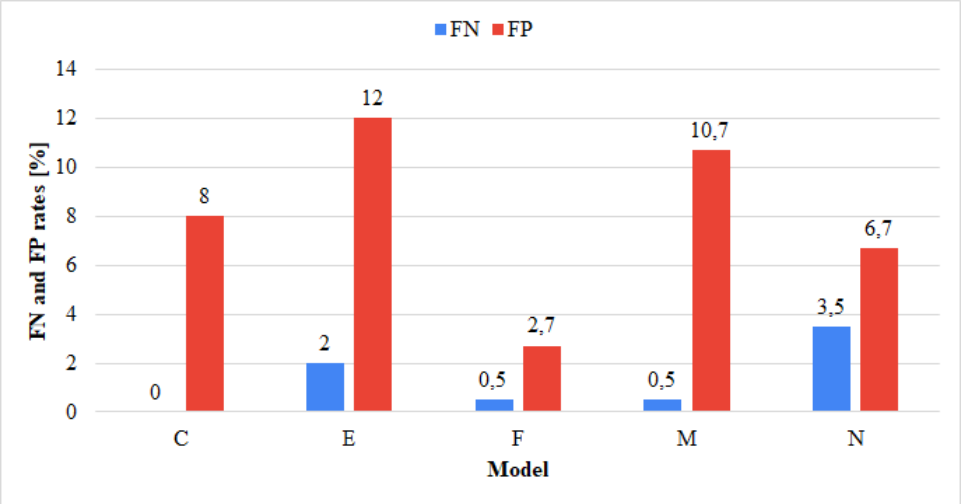
Fig 5.6 (b) shows the average false positive and false negative rate across 5 categories. There is a downward trend as more models are used. This suggest that error rates decrease as more sub-features are used. Perhaps the most interesting result was that the false positive rate decreases to absolutely 0% when all 5 features are used. Of course, during occlusion not all 5 features would be available. Category 3 models have an average false positive rate of 4.8% and Category 4 models have a false positive rate of 4%. This suggests that these two categories have the capacity to act as a recognition models that have minimal error rates.

5.4 ORL FACES DATABASE

5.4.1 Category 1



(a) True Positive and True Negative rates.

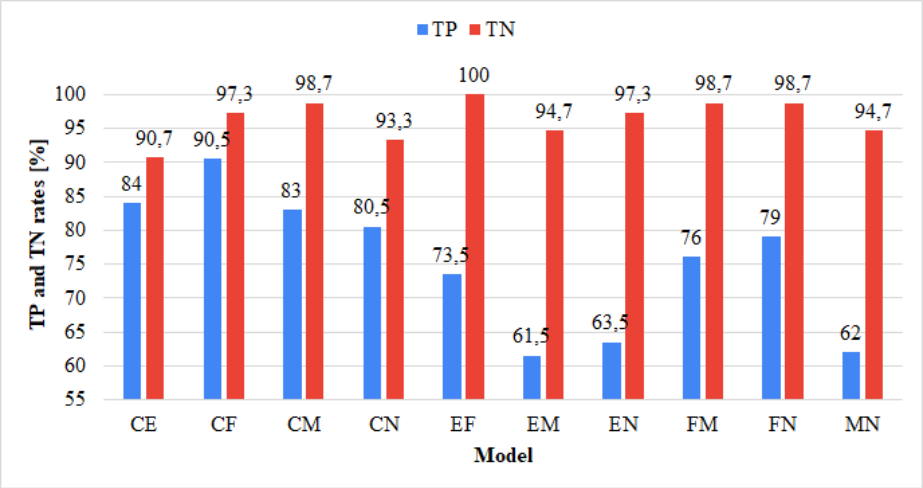


(b) False Positive and False Negative rates.

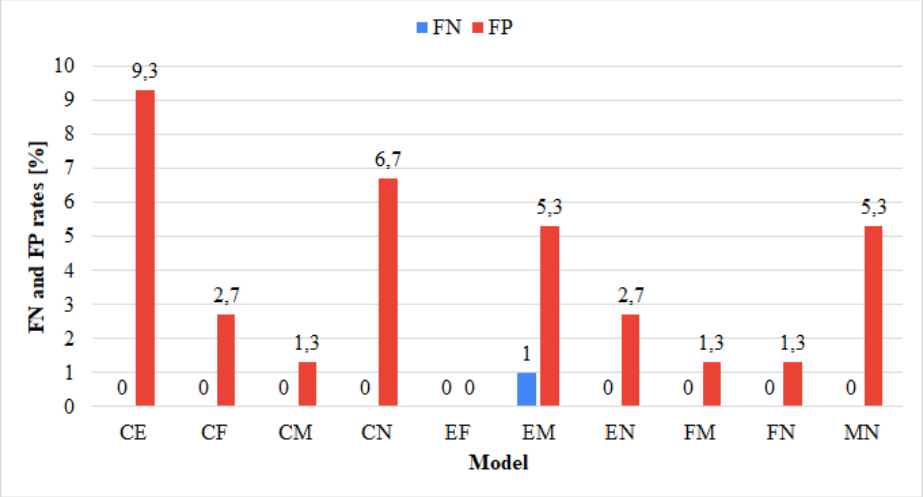
Figure 5.7. The accuracy and error rates for Category 1 models for the ORL database.

Figure 5.7 (a) shows the true positive and Figure 5.7 (b) true negative rates for Category 1 models when comparing using the ORL database. The chin and forehead models had the highest recognition rate at 72.5% and 71% respectively, while the nose model had the lowest at 33.5% and the mouth had the second lowest at 51%. The forehead and nose models had the highest true negative rates at 97.3% and 93.3% respectively. The average recognition rate was 56.1% while the average true negative rate was 91.2%. The highest false positive rate was from the eye model followed by the mouth model. The chin model had a 0% false negative rate.

5.4.2 Category 2



(a) True Positive and True Negative rates.

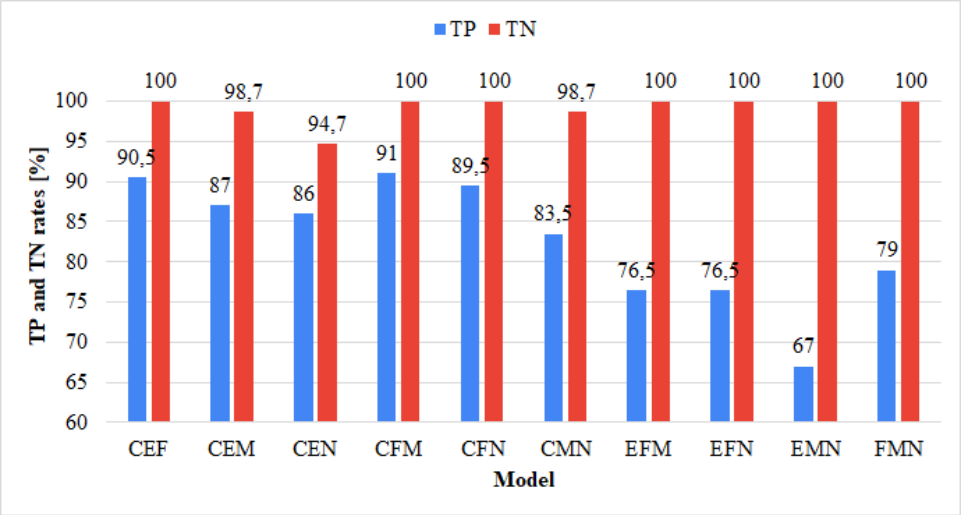


(b) False Positive and False Negative rates.

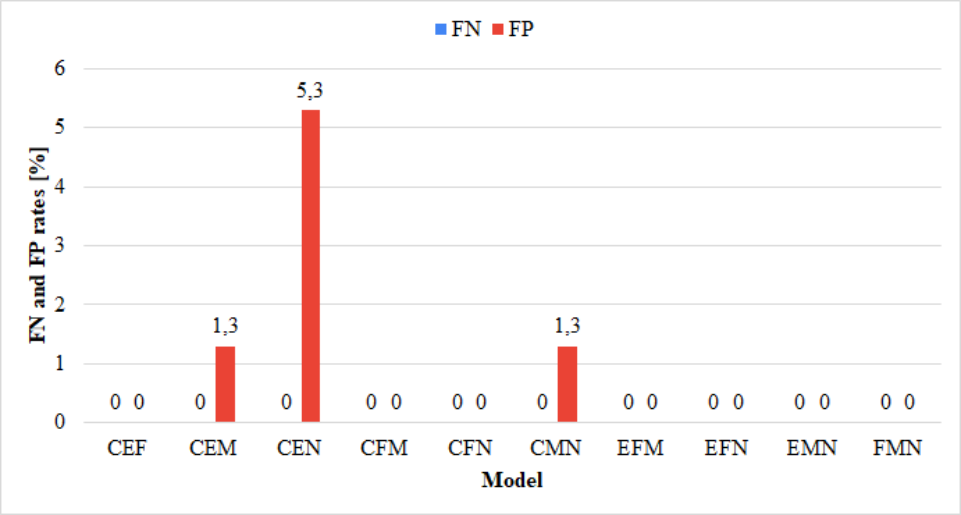
Figure 5.8. The accuracy and error rates for Category 2 models for the ORL database.

Figure 5.8 (a) shows the true positive and Figure 5.7 (b) true negative rates for the Category 2 models when comparing using the ORL database. The CF and CE models had the highest recognition rates at 90.5% and 84% respectively. The EM, MN and EN models had the lowest recognition rates at 61.5%, 62% and 63.5% respectively. The average true negative rate was 964% across all the 10 Category 2 models. The average false negative rate was 0.1% and the false positive was 3.6%. Only the EM model had a false negative rate above 0 which stood at 1%

5.4.3 Category 3



(a) True Positive and True Negative rates.

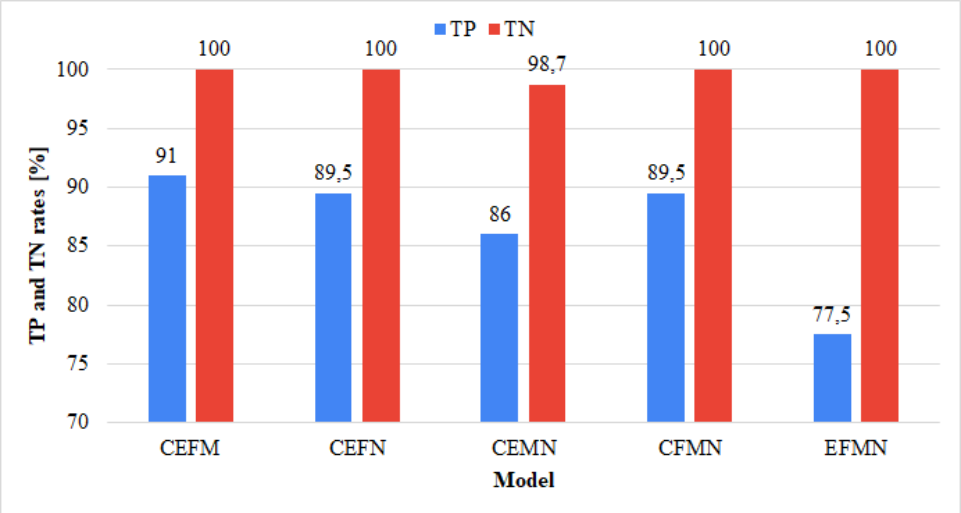


(b) False Positive and False Negative rates.

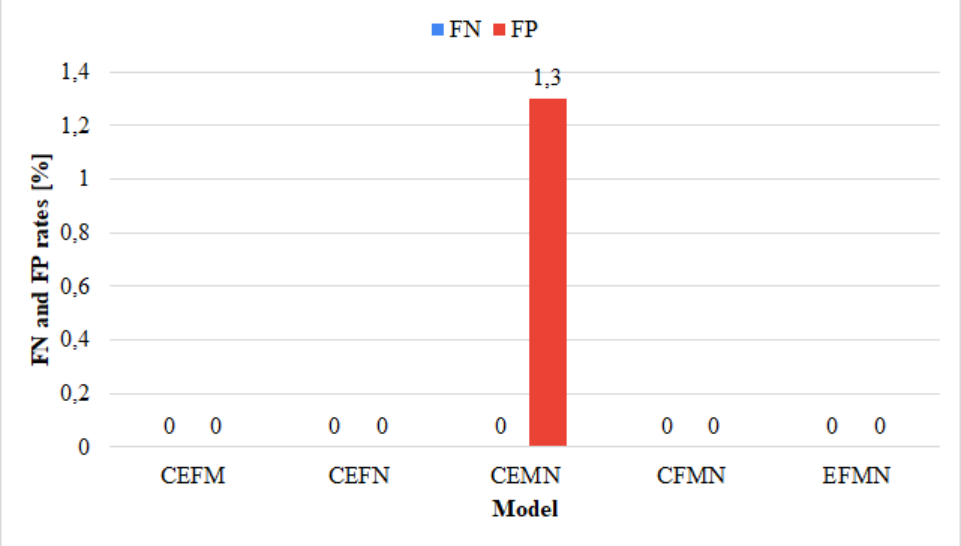
Figure 5.9. The accuracy and error rates for Category 3 models for the ORL database.

Figure 5.9 (a) shows the true positive and Figure 5.9 (b) true negative rates for Category 3 models when comparing using the ORL database. The CFM, CEF and CFM had the highest recognition rates at 91%, 90.5% and 89.5%. In contrast, the EMN had the lowest recognition rate at 67%. Only 3 models had true negative rates lower than 100%, these models were the CEN, CEM and CMN at 94.7%, 98.7% and 98.7% respectively. The false negative rate was an impressive 0%, similarly, the false positive rate was an impressive 0.79%. The CEN model single-handedly contributed to increasing the average false positive rate, it stood at 5.3%.

5.4.4 Category 4



(a) True Positive and True Negative rates.

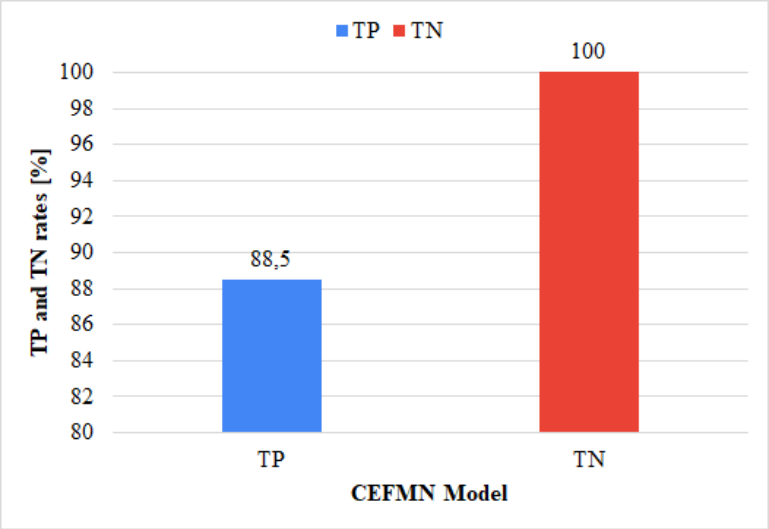


(b) False Positive and False Negative rates.

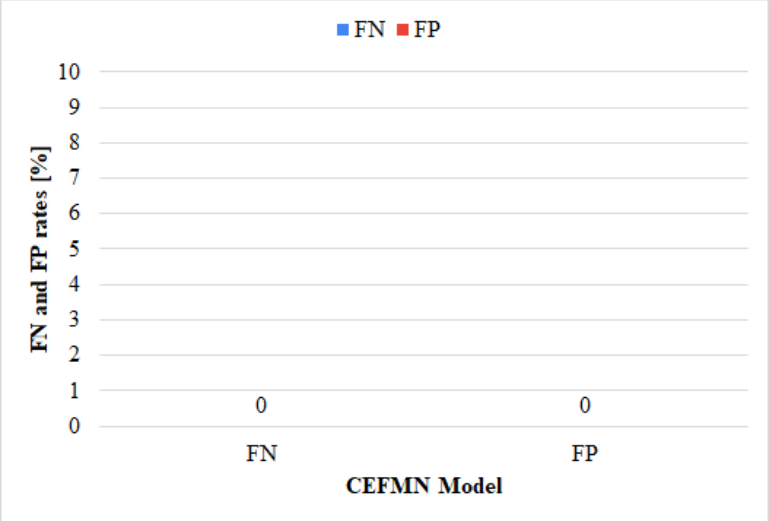
Figure 5.10. The accuracy and error rates for Category 4 models for the ORL database.

Figure 5.10 (a) shows the true positive rate and Figure 5.10 (b) true negative rate for Category 4 models when compared to the ORL database. The CEFM and CEFN had the highest recognition rates at 91% and 89.5% respectively, having said that the EFMN model had the lowest at 77.5%. All other models had a perfect true negative rate at 100% except for the CEMN model which had a rate of 98.7%. The false negative and false positive metrics were impressive as they all had 0% except for a single model, the CEMN, which had a 1.3% false positive rate.

5.4.5 Category 5



(a) True Positive and True Negative rates.

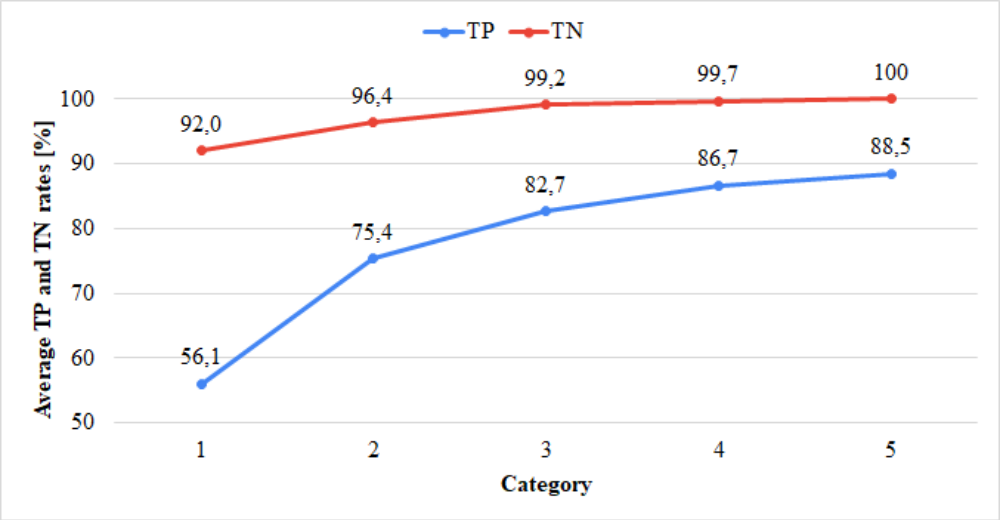


(b) False Positive and False Negative rates.

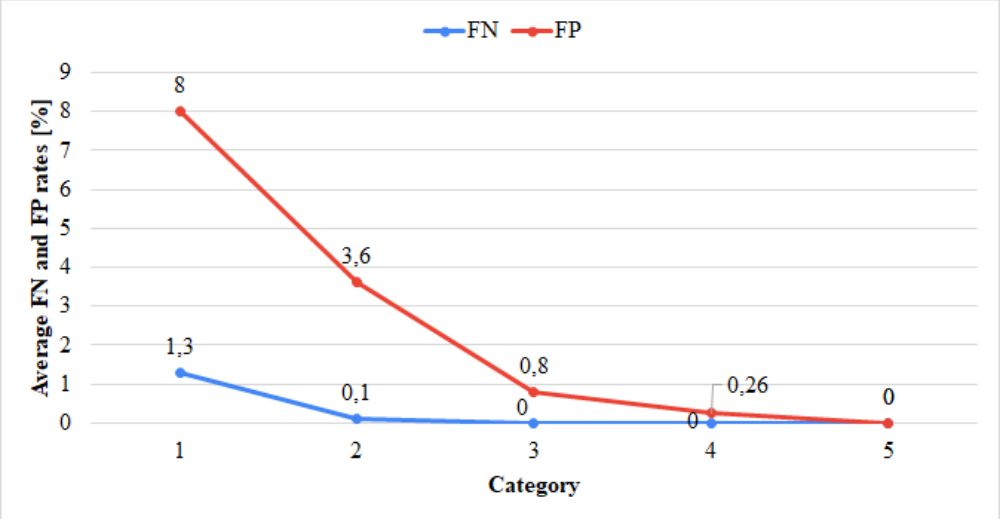
Figure 5.11. The accuracy and error rates for Category 5 models for the ORL database.

Figure 5.11 shows the accuracy and error metrics for a Category 5 model on the ORL database. The recognition rate was 88.5%, whereas the true positive rate was 100%. The error metrics were outstanding for the Category 5 model, the false negative rate and false positive rates were at an astounding 0%.

5.4.6 Discussion of results obtained on the ORL database



(a) Average true positive and average true negative rates.



(b) False Positive and False Negative rates.

Figure 5.12. The average metrics across the 5 different categories on the ORL database.

Figure 5.12 shows the average accuracy and error metrics across the five different categories on the ORL database. Category 1 models had the lowest recognition rate at 65.1% while the Category 5 model had the highest recognition rate at 88.5%. The average recognition rate was 76.9%. Only adding 1 more feature had the effect of increasing the recognition rate by 19.3% between Category 1 and Category 2 models. This shows that two features were sufficiently superior as compare to only a single feature.

On the other hand, adding a single feature from Category 4 models to a Category 5 only resulted in a 1.8% increase in the recognition rate. Looking at it qualitatively, there isn't that much difference in Category 4 and Category 5 models when it comes to recognition rates. In real world applications, when there are minimal occlusions on the face, Category 4 models can be utilised for recognition without suffering major efficiency problems.

There was an upward trend on the true positive and true negative rates when more features were added. This highlights the fact that as more information is available the better the accuracy is. The true negative rate was impressive as all the five models had true negative rates higher that 92% with an average of 97.3%. The Category 5 model remained perfect at a rate of 100%.

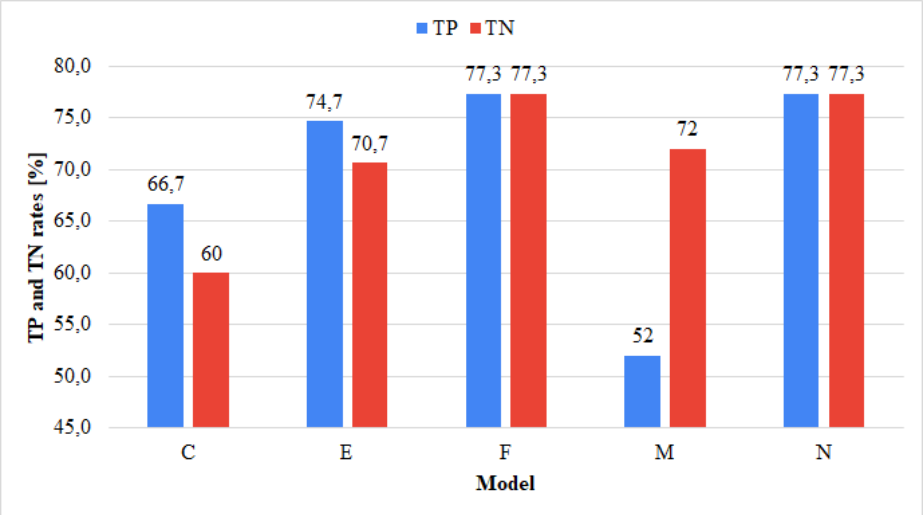
For Category 1 models the C and F models had the highest recognition rate at 72.5% and 71% respectively. The trend of having the F sub-feature having the highest recognition rates continued to the Category 3 models as well. The CFM, CEF and CFN had the highest recognition rates at 91%, 90.5% and 89.5% respectively. For Category 4 models the combination of the EFxx models proved to having higher recognition rates as well. This combination had the highest true positive rates as evidenced by the CEFM and CEFN models had 91% and 89.5% respectively.

The error metrics were equally as impressive as the true negative rate. The two error metrics had averages below 3%. The false positive rate was 2.7% while the false negative was a majestic 0.2%. This shows that not only was the segmenting and weighing technique good at recognition it was also good at minimising the errors as well. Category 1 models had the worst error rates as compared to the other four categories. The average false positive rate was 8% while the false negative rate was 1.3% for Category 1 models. It has to be highlighted that the Category 5 model had perfect error rates, both at 0%.

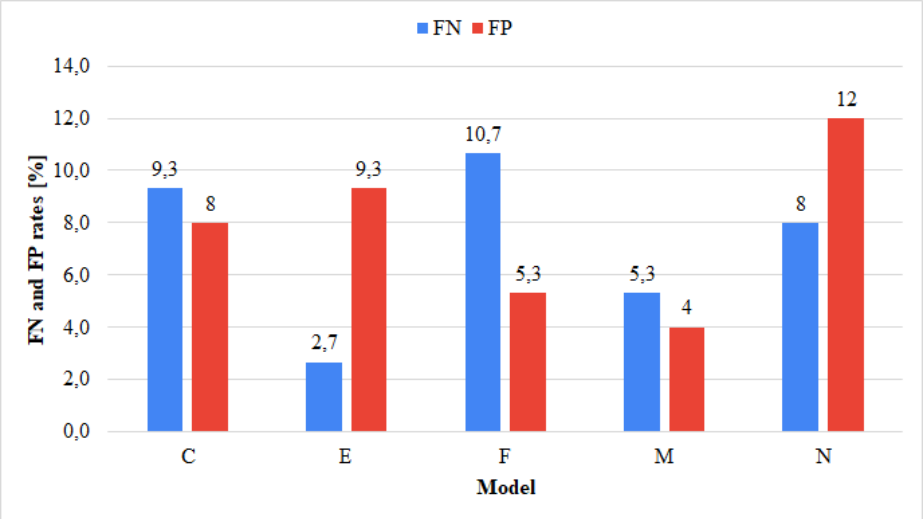
As more features were added, error rates decreased until they reached 0%. Further highlighting that, the more information that is available for the proposed technique, the better it is at minimising the errors associated with feature extraction and classification. These low error rates signify that in the real world the technique even if a person is occluded the system would be able to distinguish unknown people and would have very low misclassification errors.

5.5 YALE FACES DATABASE

5.5.1 Category 1



(a) True Positive and True Negative rates.

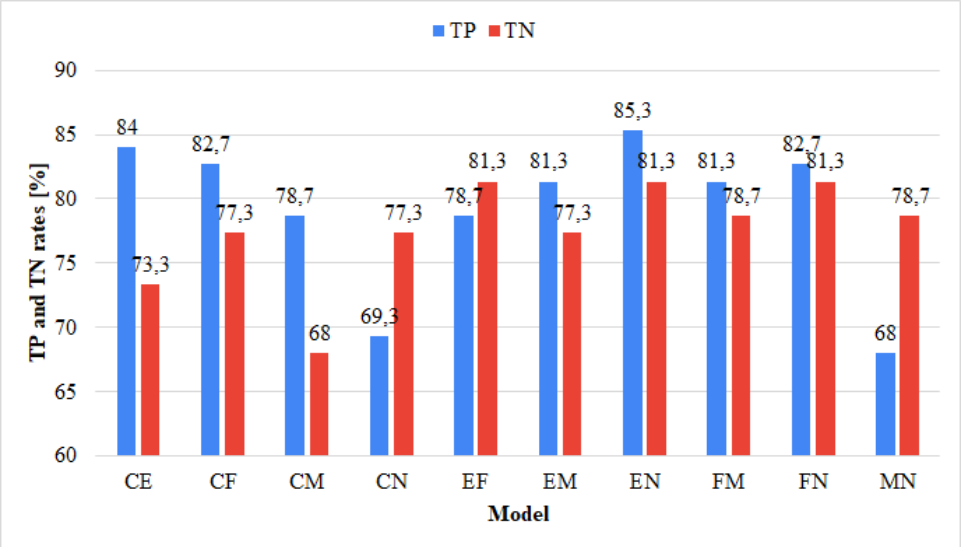


(b) False Positive and False Negative rates.

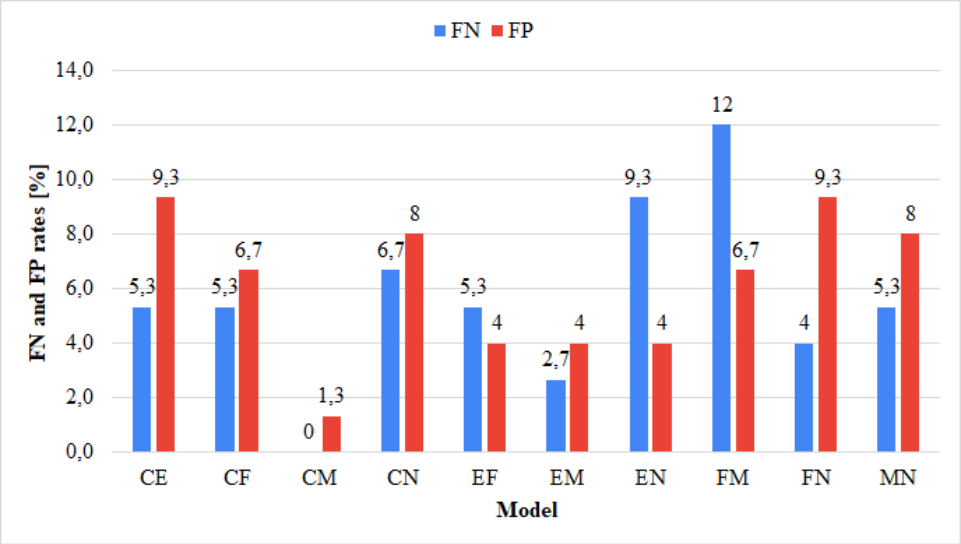
Figure 5.13. The accuracy and error rates for Category 5 models for the Yale Faces database.

Figure 5.13 (a) shows the true positive and Figure 5.13 (b) true negative rates for the Category 1 models when compared to the Yale faces database. The F and N models had the highest recognition rates and true negative rates, both at 77.3% for both models. That being said, the M model had the lowest recognition rate at lowly 52%, just a shade above splitting even. The C model had the second lowest at 66.7% for Category 1 models. The F model had the highest false negative at 10.7% while the E model had the lowest at 2.7%. The N model had the highest false positive rate at 12% while the lowest was the M model at 4%.

5.5.2 Category 2



(a) True Positive and True Negative rates.

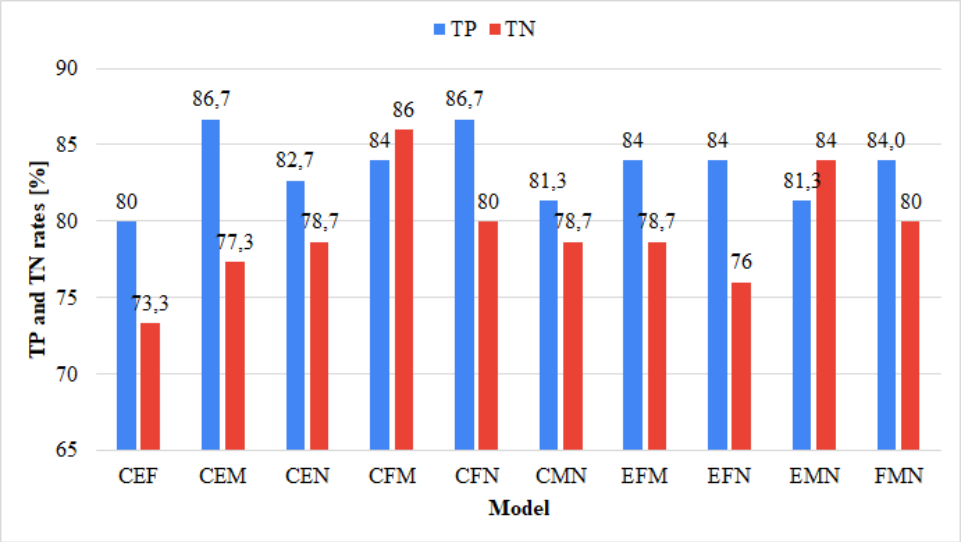


(b) False Positive and False Negative rates.

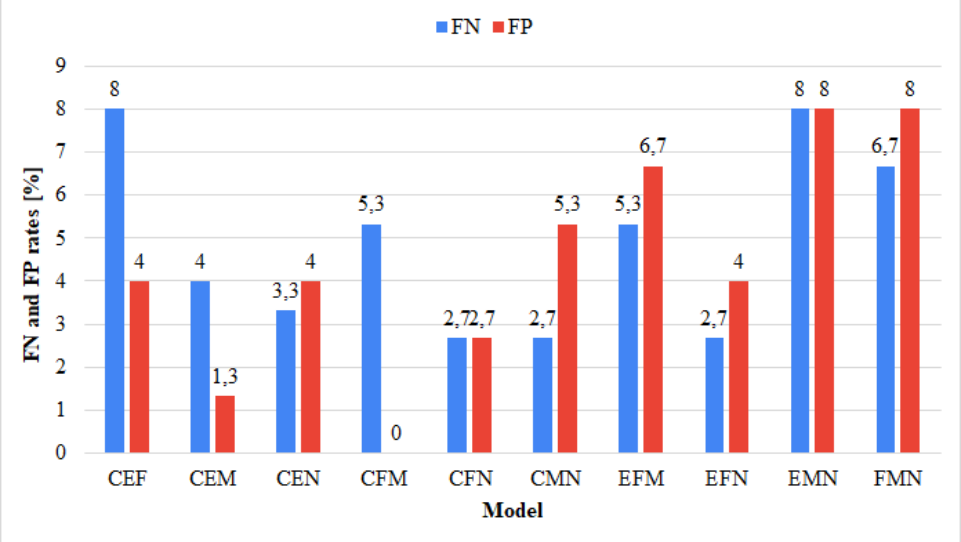
Figure 5.14. The accuracy and error rates for Category 2 models for the Yale faces database.

Figure 5.14 (a) shows the true positive and Figure 5.14 (b) true negative rates for the Category 2 models when compared to the Yale faces database. The highest recognition rates belonged to the EN, CE, CF and FN models at 85.3%, 84%, 82.7% and 82.7% respectively. The MN had the lowest true positive rate at 68%. The EF, EN and FN models were all tied for the highest true positive rate at 81.3%. The EN and FM models had the highest false negative rate at 12% and 9.3% while the CM model had the lowest at 0%. The highest false positive rate occurred on the CE model with an average of 9.3% while the lowest had only 1.3%.

5.5.3 Category 3



(a) True Positive and True Negative rates.

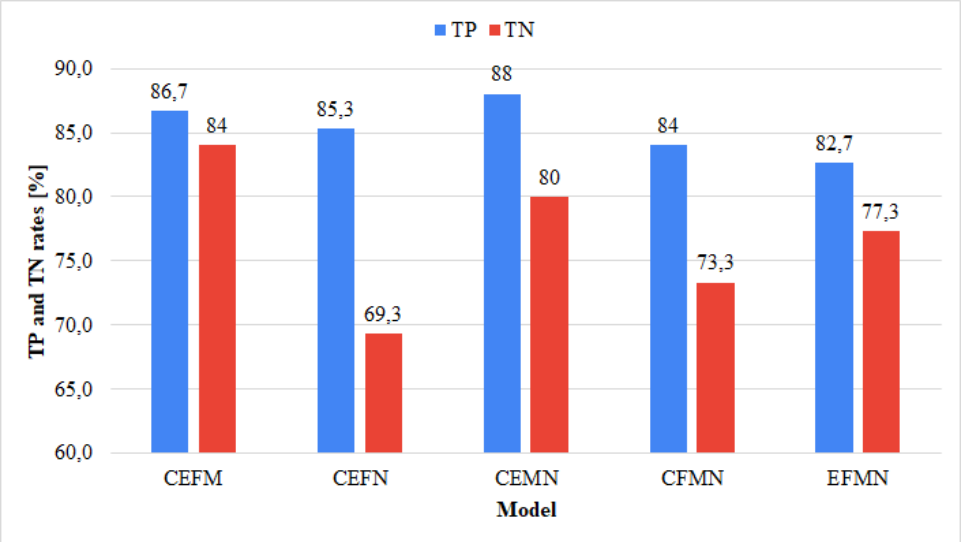


(b) False Positive and False Negative rates.

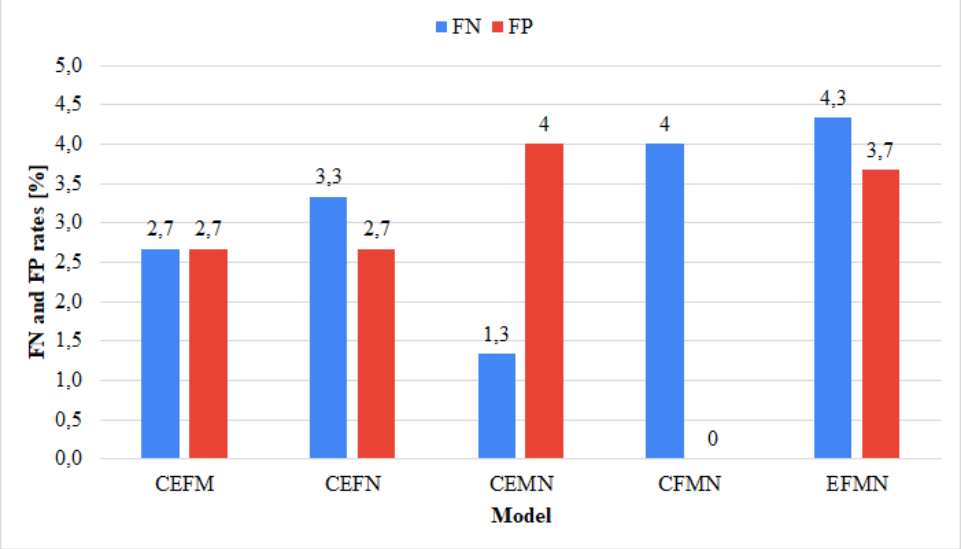
Figure 5.15. The accuracy and error rates for Category 3 models for the Yale faces database.

Figure 5.15 (a) shows the true positive and Figure 5.15 (b) true negative rates for the Category 3 models when compared to the Yale faces database. The CFN and CEM models have the highest recognition rate at 86.7% followed by the CFM, EFM, EMN and FMN all tied at 84%. The CFM model has the highest true positive rate on the other hand, the CEF has the lowest. the EMN model had the highest for both false positives and false negatives at 8%. The CFM had the lowest false positive rate at 0%.

5.5.4 Category 4



(a) True Positive and True Negative rates.

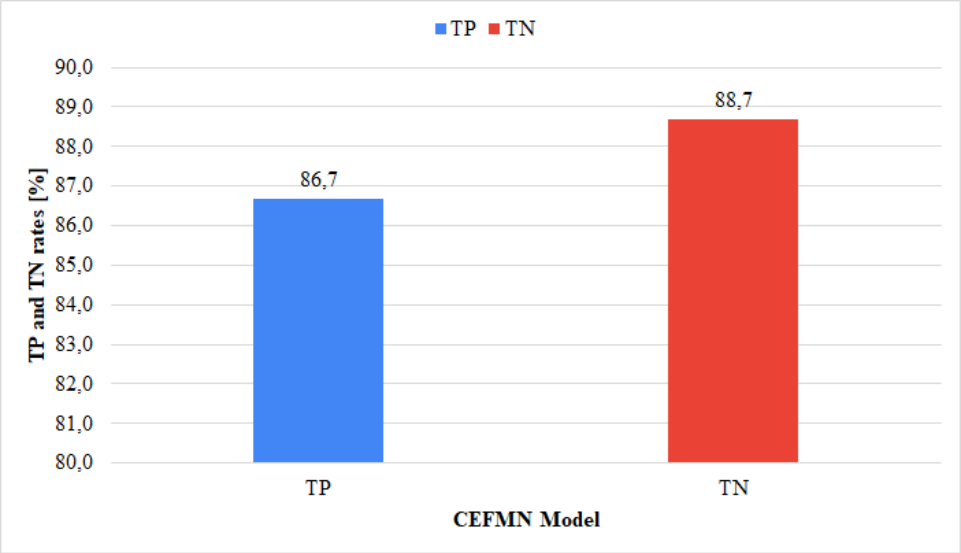


(b) False Positive and False Negative rates.

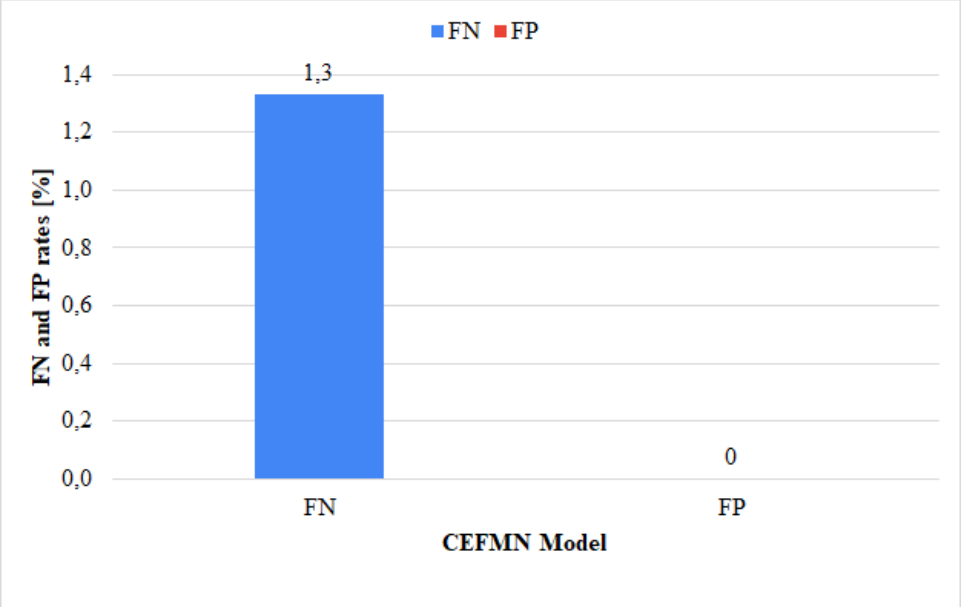
Figure 5.16. The accuracy and error rates for Category 4 models for the Yale faces database.

Figure 5.16 (a) shows the true positive and Figure 5.16 (b) true negative rates for the Category 4 models when compared to the Yale faces database. The CEMN and CEFM models had the highest recognition rates at 88% and 86.7% respectively, these models had the highest true negative rates at 80% and 84% respectively. On the other hand, the EFMN model had the true positive rate lowest at 82.3%, while the CEFN had the lowest true negative rate at 69.3%. The EFMN and CFMN models had the highest false negative rates at 4.3% and 4% respectively. The lowest false positive rate belonged to the CFMN model at 0% while the CEMN had the highest false positive rate at 4%.

5.5.5 Category 5



(a) True Positive and True Negative rates.

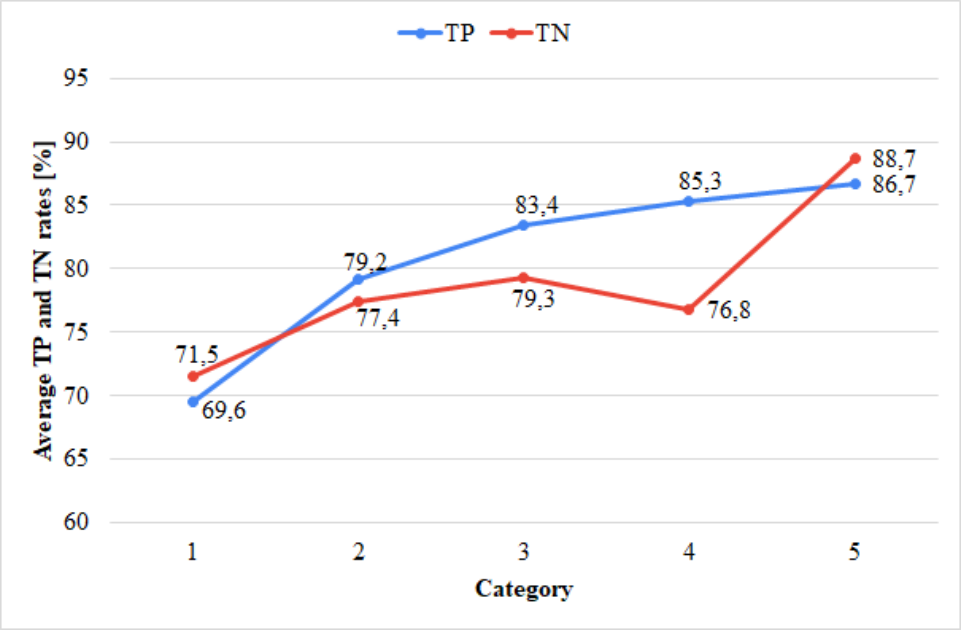


(b) False Positive and False Negative rates.

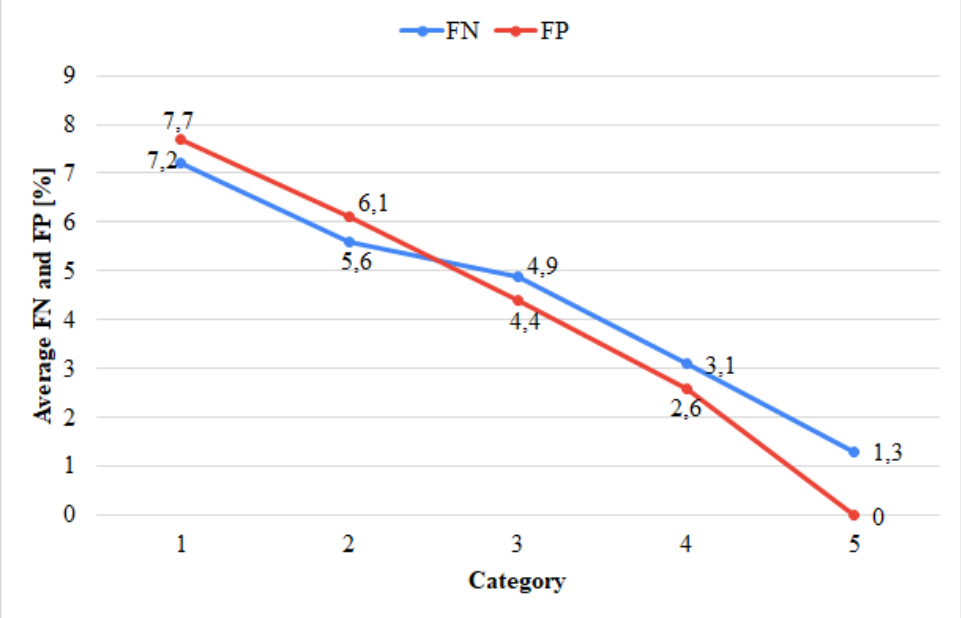
Figure 5.17. The accuracy and error rates for Category 5 models for the Yale faces database.

Figure 5.17 shows the accuracy and error metrics for a Category 5 model when compared to the Yale faces database. The true positive rate stood at 86.7% while the true negative rate was at 88.7%. The error metrics continued to be impressive as the false negative rate was just 1.3% while the false positive rate was 0%.

5.5.6 Discussion of results obtained on the Yale Faces database



(a) Average true positive and average true negative rates.



(b) False Positive and False Negative rates.

Figure 5.18. The average metrics across the 5 different categories on the Yale Faces database.

The recognition on the Yale Faces database follows a similar upward pattern as that on both the AR database and the ORL database, the recognition rate and the true negative rate both increased with the addition of more features as evidenced on Figure 5.18(a).

The lowest average recognition rate was on Category 1 models with an average of 69.6% which rose to 86.7% on the Category 5 model. The true negative had a similar trend, with Category 1 models having an average of 71.5% and the Category 5 model having the highest at 88.7%. This highlights an important aspect that the more features that are used the higher the recognition rate. There was only a 1.4% increase in the recognition rate when a Category 5 model was used instead of a Category 4 model. This is important in the sense that for efficient, a combination of 4 features can be used in the absence of the full 5 features.

The error metrics were equally as impressive, this was highlighted by the decrease in error rates as more features were used. The false positive rate decrease to a majestic 0% for a Category 5 model, similarly Category 4 models had a false positive rate of 3.1%. Thus Category 4 models could be used in scenarios where there is partial occlusion without suffering from too much decline in efficiency of the system. The average false positive rate across all categories was 4.1% while that of the false negative was 4.3%. This shows that even if strangers tried to gain access to the recognition system, only 4.1% of the times would they succeed. Conversely, registered users would only be declined 4.3% of the times. This goes to show the robustness of the facial recognition system.

Category 1 models had an average recognition rate of 69.6%, Category 2 models had 79.2%. Category 1 models had a fairly high recognition rate while only utilising a single feature. This is in stark contrast to the Category 1 models on the ORL database which had a measly 56%. Whenever the F feature was a sub-feature to other various models it followed a similar trend as that seen on the AR and ORL databases. In Category 1 models it had the highest recognition rate, while the mouth had the lowest recognition rate. This trend continued to Category 2 models as it had three of the top five models with the highest recognition rates. Bare in mind, although there are a total of ten Category 2 models, the CF, FN and FM models had the highest rates at 85.3%, 82.7% and 81.3% respectively.

Category 3 models with the F sub-feature continued to dominate as well. Category 3 models had a TP of 83.4%. The CFN model had the highest recognition rate, and the F sub-feature was a component of 5 of the top 6 models with the highest recognition rate. The models were the CFN, CFM, EFM, EFN and the FMN models. This shows that models with the F sub-feature should always be considered when building a recognition system that is capable of recognition under partial occlusion.

When it came to Category 4 models, the CEMN had the highest TP rate, while the CEFM had the second highest. Category 4 models were had a TP of 85.3%. The CExx models performed surprisingly well. This can be attributed the large variations of the chin, some subjects had different types of beards, and various emotions like smiling or grinning, which led to a larger difference between subjects within the Yale Faces database.

5.6 AVERAGE METRICS ACROSS ALL 3 DATABASES

5.6.1 Feature specific effective recognition rates (eTP)

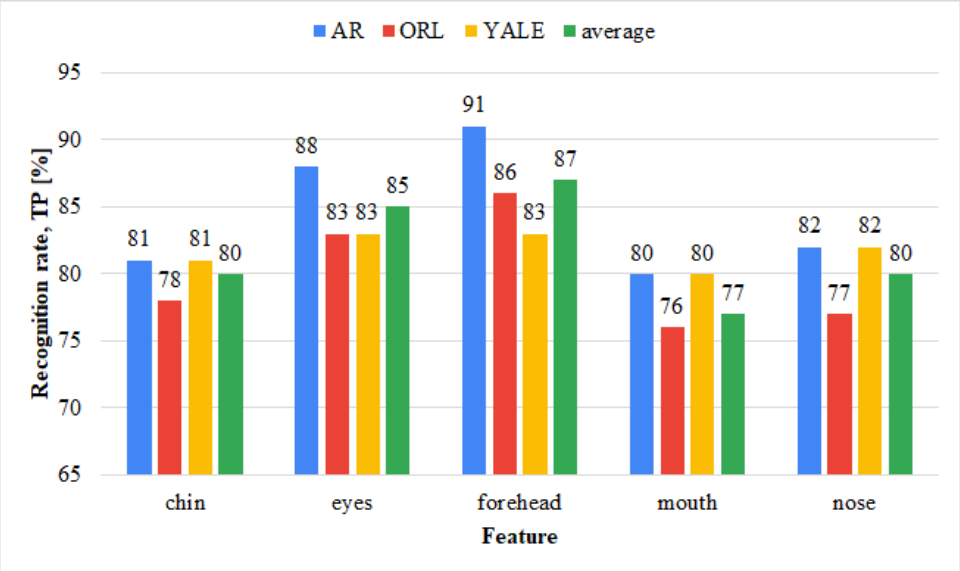


Figure 5.19. The recognition rate per feature on the different databases.

Figure 5.19 shows the average recognition rate across all databases for all combinations that contained each single feature. This means that 15 models were used to calculate the average of any single feature across all three databases. For instance, to calculate the average recognition rate of the C (chin) sub-feature, the following models were utilised; the CE, CF, CM, CN, CE, CEM, CEN, CFM, CFN, CMN, CEFM CEFN, CEMN and CFMN models were used to calculate the effective recognition rate. This pattern was repeated for the other four models in order to assess which regions offered the highest recognition rates.

Analysing these individual sub-features highlighted a trend that was seen when analysing the categories of individual databases. The forehead sub-feature had the highest, which was an astonishing effective recognition rate of 87%. It was closely followed by the eye (E) sub-feature with an average rate of 85%. Perhaps the most damning sign is that the mouth region had the least recognition rate at 77%. The chin and the nose were both tied at 80% each. This analysis was telling, the forehead was surprisingly the feature which offered the most discrimination. This meant that when parts of the face were occluded, the forehead could be used to ascertain an individual's identity, by utilising models which contained the forehead feature. The high accuracy of the forehead region can also be explained by the fact that the eyebrow region and the hair region was considered as part of the forehead region. The variation in the shapes of the eyebrows, the eyebrow orientation, and different hair styles led to vast differences between subjects. Some subjects had weaves on, others were bald, others were balding, others had

thick hair while others had thin hair. These variations were large enough to have a major bearing on which feature had the highest recognition rate.

The eyes, as expected, scored high as well. It had been largely speculated that the eyes, nose and mouth offered the most discrimination when recognition a person. This notion was partially disproved when utilising an image segmentation and weighing technique. This technique showed that the mouth actually had less discrimination as compared to the chin feature. The chin having a higher recognition rate can be attributed to this region as having vastly more variations as compares to the mouth. Subjects in these databases differed in their emotions. This meant that if someone was smiling, grinning, sad, pouting and ecstatic the chin region was affected by these models. This led to non-uniformity around this region, hence greater variationsoccured. Different subjects had different types of facial hair on their chin regions, this led to more variations.

5.6.2 Advanced accuracy and error metrics

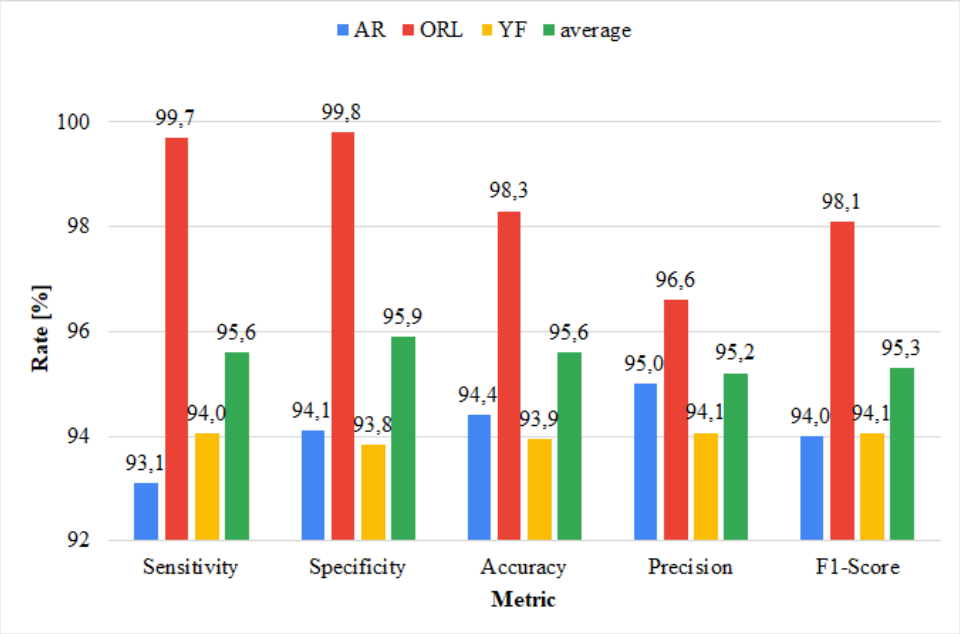


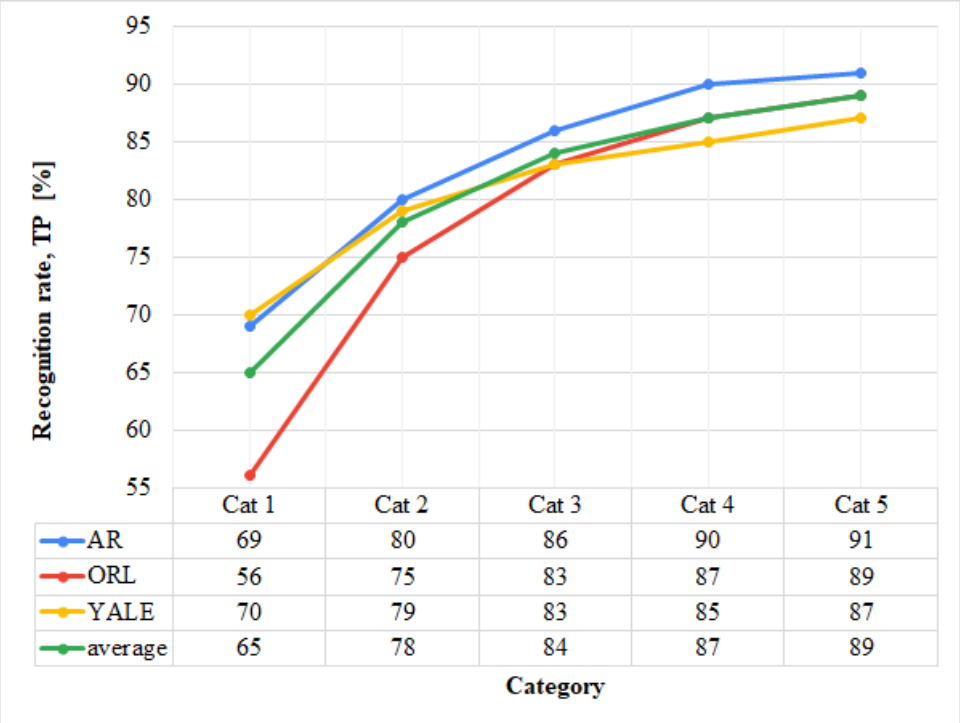
Figure 5.20. The accuracy metrics and error metrics on the three databases used.

Figure 5.20 shows the advanced accuracy and error metrics. The technique yielded an overall accuracy of 95.6%. The accuracy took into account the TP and TN values. Due to the average TN being extremely high this meant the overall accuracy was also high. The technique could discern which subjects were registered and which ones were unregistered. Since it was also important to not only evaluate the TP and TN rates, but also how well it performed at classifying data which was labelled as negative. The overall precision was 95.2%. This was a measure of how precise the model was at predicting the positives from the actual positives. This is used to measure if the cost of false positives was high. A lower precision implies that unregistered users would be able to fool the recognition

system, while a high precision signifies that the model was fully equipped to deal with unregistered users and was able to classify them as true negatives. Similarly, the sensitivity also known as recall, measured the cost of false negatives. It stood at 95.6%, signalling that the model did not suffer from misclassifying registered uses as unregistered users. The specificity measured how well the model classifies actual unregistered subjects as unregistered subjects. This measure was an impressive 96%.

Finally, the F1-Score measured the balance between precision and recall. Since there was an uneven class distribution between the actual true positives and actual negatives, the F1-Score took into account this weighed relationship. This F1-Score was 95.3%, was close to 100% meaning the technique was extremely good at not only identifying the actual positives, but also the actual negatives.

5.6.3 Average recognition rate (TP)



	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
AR	69	80	86	90	91
ORL	56	75	83	87	89
YALE	70	79	83	85	87
average	65	78	84	87	89

Figure 5.21. The average recognition rate on the 5 categories.

Figure 5.21 shows the recognition rate across on individual databases, as well as the average across all databases. The average recognition rate for Category 1 models across all databases was 65%, for Category 2 models it was 78%, for Category 3 models it was 84%, for Category 4 models it was 87% and finally for a Category 5 model it was 89%. Using only 1 feature was sub-optimal, there was an increase of 13% when another sub-feature was added for recognition. This was the biggest jump between categories. There was a successive 6% increase, then a 3% increase and finally a 2% increase to the final Category. There increases show that it is feasible to run a Category 3 or 4 model for recognition as there was only a 5% and 2% difference from the Category 5 model. It is wise though to run a Category 4 model, when a subject's face is only partially occluded and only run a Category 3 model when a subject's face is heavily occluded.

5.6.4 Average false negative rate (FN)

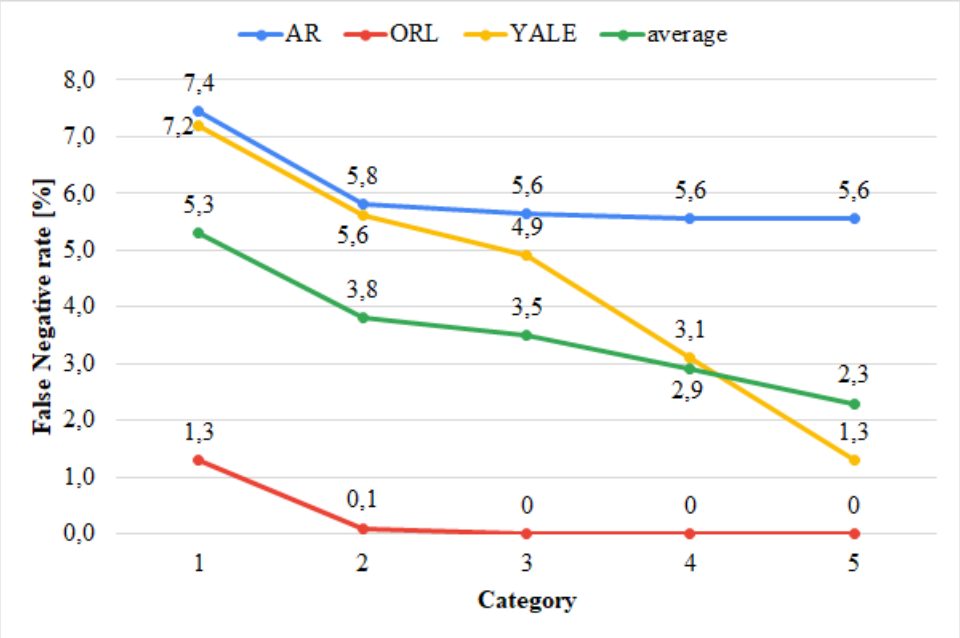


Figure 5.22. The average false negative rate on the 5 categories.

Figure 5.22 shows the average false negative rate ad the five categories of the technique across all databases. The downward trend is quite noticeable as the Category 1 model had a 5.3% false negative rate while the Category 5 model had a false negative rate of 2.3%. This shows that the recognition system only suffered with misclassifying registered uses to a very low extent. Category 1 models had the worst rate since only a single feature was utilised for recognition, this is logical as less information was available to the recognition system.

5.6.5 Average false positive rate (FP)

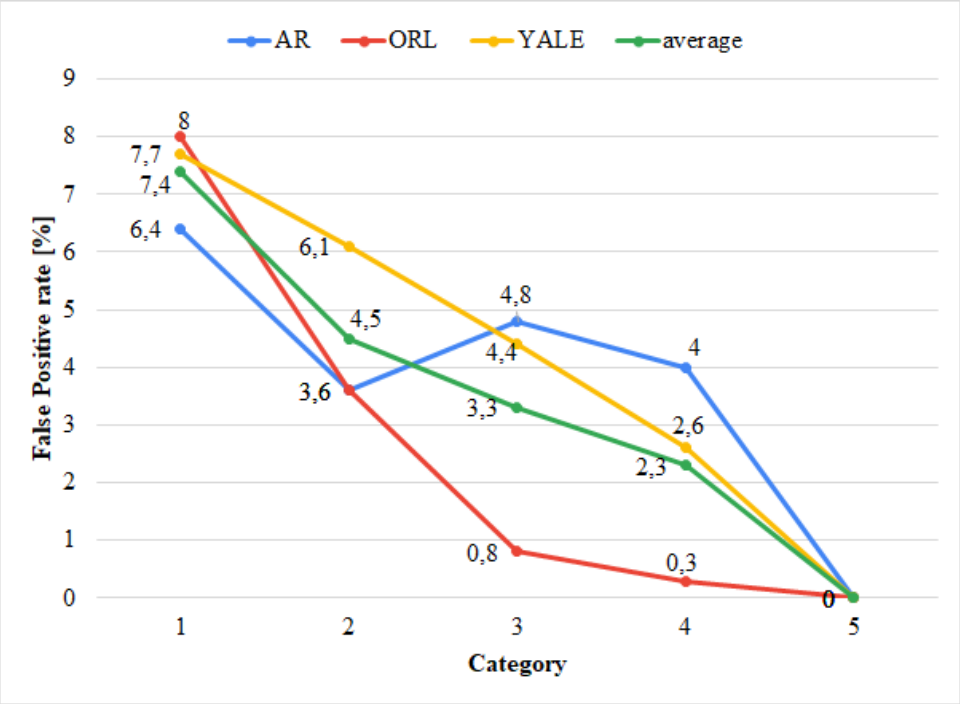


Figure 5.23. The average false positive rate on the 5 categories.

Figure 5.23 shows the average false positive rate of the face recognition system across five different categories. The false positive rate decreased from 7.7% for Category 1 models down to an astounding 0% for Category 5 models. This meant that, on average, for a Category 5 model there were no unregistered subjects that were misclassified as registered. This is an impressive feat, as it shows that rogue elements can not gain access to an area secured by the face recognition system using a Category 5 model.

5.7 COMPARISON WITH OTHER TECHNIQUES

For there to be an objective discussion on the performance of the proposed technique, that was implemented in this dissertation, there is a need to compare the key results obtained against other algorithms in literature and also those algorithms that were implemented in Chapter 3. This section aims to objectively seek that end goal.

5.7.1 Recognition rate

This section presents the results from the experiments that were carried out to ascertain the recognition rate of the proposed algorithm against other methods in literature. Table 5.1 shows the comparison of the recognition rates obtained from the proposed technique, the implemented methods and various other methods in the literature. All these comparisons were done on three readily available databases; AR database, ORL database and the Yale Faces database.

Table 5.1. A comparison of recognition rates.

Method	Recognition rate			
	AR (%)	ORL (%)	Yale (%)	other (%)
KPCA [140]	83	-	-	-
LGBP [141]	80	-	-	-
PSVM [142]	82	-	-	-
PUM [143]	80	-	-	-
SRC [144]	84	-	-	-
PUDMNN [145]	82	84	-	-
3DCNN [146]	-	94	-	-
FHMM [147]	-	87	-	-
Gabor-SVD [148]	-	75	-	-
WPCA [149]	-	86	-	88 (UMIST)
2DPCA-ICA [150]	-	83	79	-
ALBP [151]	-	-	62	68 (JAFFE)
DLBP [151]	-	-	56	57 (JAFFE)
LBP [151]	-	-	61	60 (JAFFE)
LDP [151]	-	-	71	73 (JAFFE)
DCT (implemented)	77	89	88	-
ICA (implemented)	79	85	83	-
NMF (implemented)	54	74	77	-
SRC (implemented)	75	79	81	-
Proposed technique cat. 1	69	56	70	-
Proposed technique cat. 2	80	75	79	-
Proposed technique cat. 3	86	83	83	-
Proposed technique cat. 4	90	87	85	-
Proposed technique cat. 5	91	89	87	-

Table 5.2. Abbreviations used in Table 5.1 and their full meanings.

Method abbrev.	Name
PUM	Posterior union model
KPCA	Kernel PCA
LGBP	Local Garbor binary patterns
SRC	Sparse-representation based classification
PSVM	Partial support vector machine
WPCA	Wavelet principal component analysis
FHMM	Fuzzy hidden Markov models
Garbor-SVD	Garbor singular value decomposition
PUDBNN	Posterior union model decision-based neural network
3DCNN	Three-dimensional convolutional neural network
2DPCA-ICA	Two-dimensional PCA-ICA
LDP	Local derivative pattern
ALBP	Advanced local binary pattern
LBP	Local binary pattern
DLBP	Dominant local binary pattern
JAFFE	Japanese female facial expression database
UMIST	University of Manchester Institute of Science and Technology database

In Table 5.1, when considering the AR database, the proposed technique accounted for three of the top five methods with the highest recognition rates. The top three being proposed technique when using a Category 5 model, then Category 4 models and then Category 3 models at 91%, 90% and 86% respectively. The SRC method and the KPCA methods rounds up the top 5. The proposed technique for Category 1 models at 69%, Category 2 models at 80%, SRC method at 80%, LGBP at 80% and PSVM at 82% had the lowest recognition rates. This shows that on the database with occlusion the proposed technique outperforms most of the methods in literature that are captured on this table. This makes this technique suitable for situations where occlusions are present. The other methods listed on the table performed either relatively well or below average. The proposed technique for both Category 3 and 4 models outperforms the KPCA at 83%, LGBP at 80%, PSVM at 82%, PUM at 80%, SRC at 84% and the PUDMNN at 82% methods taken from literature.

The proposed technique was compared with the four methods, which were implemented sorely for comparison purposes. These implemented methods were DCT at 77%, implemented ICA at 79%, implemented NMF at 54% and implemented SRC at 75%. The proposed technique using Category 3 models had an average rate of 86%, Category 4 models had an average rate of 90% and a Category 5 model had an average rate of 91%. The proposed technique outperformed these methods, regardless of the fact that this technique had less facial information to work with.

When comparing the proposed technique against other methods in literature on the ORL database, the Category 5 model at 89% and Category 4 models at 87% were in the top 5, being second and third.

The highest recognition rate was for the 3DCNN at 94% for this database. The method which had the lowest recognition rate was the proposed technique using Category 1 models at 56%, followed by the proposed technique at 75%, followed by Gabor-SVD at 75% and finally by the 2DPCA-ICA at 83%. When comparing the proposed technique to the implemented methods used for comparison purposes, Category 4 models at 87% outperformed the implemented ICA at 85%, implemented NMF at 74% and implemented SRC method at 79%. When analysing Table 5.1, the proposed technique, using Category 4 models, outperformed the Gabor-SVD at 75%, 2DPCA-ICA at 83%, WPCA at86% and the PUDMNN method at 84% that are found in literature. This is a feat to be celebrated as Category 4 models only use limited features of the face and do not use the entire face as compared to the stated methods.

When using the Yale Faces as the database for comparisons, the proposed technique using Category 3, 4 and 5 models all outperformed the 2DPCA-ICA at 79%, ALBP at 62%, DLBP at 56%, LBP at 56% and LDBP at 71% methods taken straight from literature. The highest recognition rate belonged to the proposed technique using Category 5 models at 87%. The method which had the lowest recognition rate was the DLBP at 56%, followed by the LBP at 61% and the ALBP rounded out the bottom three at 62%. When comparing the proposed technique and implemented methods, which were used for uniformity sake, only the DCT at 88% outperformed the proposed method using Category 4 at 85% and 5 at87% models. Other than that these two categories outperformed the remaining three implemented methods which were the implemented ICA at 83%, implemented NMF at 77% and implemented SRC at 81%.

5.7.2 False acceptance rate (FAR)

Table 5.3 shows the comparison of the false acceptance rates between the proposed technique, the implemented methods and the methods taken directly from literature, for comparison purposes. In face recognition, FAR is a measure of the percentage of all unauthorised subjects that are incorrectly accepted. The proposed technique and the other methods are compared against three databases; AR database, ORL database and the Yale Faces database.

When these methods are compared against the AR database, the proposed technique yielded four of the top five best rates. These four were the Category 5 model at 0%, followed by Category 2 models at 3.6%, then Category 4 models at 4% and the fourth one being the Category 3 models at 4.8%. The implemented NMF and the implemented SRC had the worst performance at 11.6% and 8.3% respectively.

When it came to comparisons against the ORL database, the trend of the proposed technique outperforming other methods in literature continued. The proposed technique had four of the top five

Table 5.3. A comparison of false acceptance rates.

Method	False acceptance rate		
	AR (%)	ORL (%)	Yale (%)
DCT (implemented)	5.8	4	6
ICA (implemented)	5.8	1.7	3.9
NMF (implemented)	11.6	7.1	9.6
SRC (implemented)	8.3	4.6	6.0
PDBNN [152]	-	9.24	-
DT-CWT [153]	-	18.5	-
ZDCT [154]	-	10.7	-
BDPCA+LDA [155]	-	-	7.4
Proposed technique cat. 1	6.4	8	7.7
Proposed technique cat. 2	3.6	3.6	6.1
Proposed technique cat. 3	4.8	0.8	4.4
Proposed technique cat. 4	4	0.3	2.6
Proposed technique cat. 5	0	0	0

Table 5.4. Abbreviations used in Table 5.3 and Table 5.5 and their full meanings.

Method abbrev.	Name
BDPCA + LDA	Bi-directional PCA plus LDA
DT-CWT	Dual-tree complex wavelet transform
PDBNN	Probabilistic decision-based neural network
ZDCT	Zoned discrete cosine transform
2DICA	Two-dimensional independent component analysis
RBFN	Radial basis function network

false acceptance rates. The proposed technique using a Category 5 model had the best rate at 0% again, then followed by Category 4 models at 0.3%, then Category 3 models at 0.8% then finally the Category 2 models at 3.6%. These four models outperformed the PDBNN at 9.2%, ZDCT at 10.7% and finally the DT-CWT method at 18.5%. This DT-CWT method had the worst false acceptance rate at 18.5%. In comparison to the implemented methods, the implemented DCT had an average FAR of 4%, implemented ICA had a FAR of 1.7% and the implemented SRC had a FAR of 4.6%. These implemented methods only managed to outperform Category 1 models which had an average rate of 8%.

Table 5.3 also shows the comparisons against the Yale faces database. For this database, the proposed technique using a Category 5 model performed the best at 0%, followed by Category 4 models at 2.6%. The dominance of the proposed technique continued to show even while using this database which has major variations. The Category 4 and 5 models outshone all other methods, including the BDPCA+LDA method at a dismal rate of 7.4%.

5.7.3 False rejection rate (FRR)

Table 5.5. A comparison of false rejection rates.

Method	False rejection rate		
	AR (%)	ORL (%)	Yale (%)
DCT (implemented)	8.3	4.9	7
ICA (implemented)	7.1	1.5	6
NMF (implemented)	9.4	6.8	7.1
SRC (implemented)	6.6	4.7	6
PDBNN [152]	-	7.2	-
2DICA [156]	0.47	0.45	0.012
DT-CWT [157]	-	1.5	-
RBFN [158]	-	0.26	0.4
ZDCT [154]	-	13.5	-
BDPCA+LDA [155]	-	7.4	-
Proposed technique cat. 1	7.4	1.3	7.2
Proposed technique cat. 2	5.8	0.1	5.6
Proposed technique cat. 3	5.6	0	4.9
Proposed technique cat. 4	5.6	0	3.1
Proposed technique cat. 5	5.6	0	1.3

The final error metric that was used to assess the holistic performance of the proposed technique was the false rejection rate. In face recognition, FRR is a measure of the percentage of all authorised subjects who are incorrectly rejected. Table 5.5 shows the comparison of the results obtained for false rejection rates. All these methods were compared using different databases.

When it comes to the AD database the 2DICA had the best false rejection rate at 0.47%. The four next best methods were the proposed technique using a Category 5 model, Category 4 models, Category 3 models all tied at 5.6 then lastly the proposed technique using Category 2 models at 5.8%. The worst performing was the implemented SRC at 9.4% followed by the implemented DCT at 8.3%.

When it comes to the ORL database, the proposed technique had perfect false acceptance rates for 3 of its models which were the Category 5 model, Category 4 models and Category 3 models all had an astounding 0% false acceptance rate, save for the Category 2 models which had a 0.1% rate. These four techniques outperformed all the tabulated methods taken from literature which were the RBFN at 0.26%, DT-CWT at 1.5%, the 2DICA at 0.45%, the DBPAC+LDA at 7.4% and the ZDCT at 13.5%.

Surprisingly, when testing on the Yale Faces database, the proposed technique was outperformed by 2 methods from literature. These methods were the RBFN at 0.4% and the 2DICA at 0.012% which was also the best performer for this particular database. The worst performers were the implemented DCT

at 7%, the implemented NMF at 7.1% and finally the proposed technique utilising Category 1 models at 7.2%.

5.8 CONCLUDING REMARKS

This chapter presented and discussed the results of the proposed technique, results from the implemented methods and a comparison with results from literature. These results were compared against those found in literature. All experiments were carried out on three readily available databases, namely the AR database, ORL database and the Yale Faces database. In order to obtain the results, the proposed technique utilised an image segmentation and weighing technique that was compared against the three databases. Results were obtained by a systematic approach that made use of five-fold cross validation. The training set and testing set were split into a ratio of 0.7:0.3.

Section 5.3 - 5.5, focused on the results obtained from the proposed technique, and each section reported on the five different Category's accuracy and error metrics. These metrics were the TP, TN, FP and FN.

Section 5.6 then focused on the average accuracy and error metrics across all the database. This section presented the average recognition rates of the five features that were extracted from the face which were the chin, eyes, forehead, mouth and nose. This section also presented advanced accuracy and error metrics that established the performance of the proposed technique's ability to measure the cost of false positives and false negatives. These advanced metrics were the sensitivity, specificity, accuracy, precision and the F1-score.

For there to be an objective discussion about its performance, Section 5.7 focused on the comparisons between the proposed technique against other face recognition algorithms and methods taken directly from literature for comparison purposes. This section compared the recognition rates, false acceptance rates and false rejection rates.

CHAPTER 6 CONCLUSION

6.1 SUMMARY OF WORK CONDUCTED

In this dissertation, the problem of partial occlusions on faces was tackled. Humans can automatically recognise a person's face when it is under minimal to extreme occlusions. This problem of face recognition under partial occlusion is quite different when it comes to computer aided algorithms. Since the face would be partially occluded, there is less information to work with during the recognition process. Partial occlusions, such as sunglasses and scarfs, lead to loss of information on the eyes, nose, mouth or chin. This dissertation focused on building a robust face recognition system that utilised this limited information in order to efficiently recognise a person's face. Not all regions of the face contribute equally to the recognition process, some regions have been theorised to contribute less and some have been theorised to contribute most, if not all, to the recognition process. Eyes and the nose are traditionally considered to be the regions that contribute more, and the chin and forehead are the regions that have been theorised to contribute less.

In order to obtain results that could both be compared against other methods in literature, there was a need to conduct experiments systematically. First of all images in the three databases used were standardised to be the same pixel width and height. A resolution of 120 x 120 pixel was chosen. Feature detection was then carried out in order to detect the five key features under investigation. This dissertation aimed to qualitatively and quantitatively investigate which regions of the face contribute the most. It also aimed to assign weights to regions of the face that contributed mostly to face recognition. Some features of the face are essential in recognising a person's identity, so this dissertation put forward a function that assigns weights to the features of the face according to their importance. Thus it makes use of the Gaussian function that has five centres, centred on the left eye, the right eye, the centre of brows, the nose and the mouth. Weights were assigned to the features of the face starting from these centres and emanating to the rest of the face. This effectively assigned higher weights to those features closer to the centres and lesser weights to those further away.

Since the other objective was to investigate which sub-regions contributed the most when it comes to recognition, this dissertation made use of the image segmentation technique. An image was segmented

into five prominent regions which contained the chin (C), eyes (E), forehead (F), mouth (M) and nose (N). This meant that the cheek and the ear sub-regions were discarded. These five selected features were used to construct 31 models, as tabulated on Table 4.1, that contained all possible combinations these features. For example, the chin feature would have the following models; C, CE, CF, CM, CN, CE, CEM,CEN, CFM, CFN, CMN, CEFM CEFN, CEMN, CFMN and CEFMN. This was done to all the five features in order to construct their own models for training and testing purposes. These models were divided into five categories, Category 1 models contained single feature models only, Category 2 models contained models with two features. The division of Category 3, 4 and 5 models was done similarly.

Since the AR database had occlusion, some of the features like the eyes, mouth and chin were not detected because there were either sunglasses or scarfs covering these features. If sunglasses were present, there would only be 4 features selected. This led to the construction of a Category 4 model for this particular image. On the other hand, if a scarf was present this led to two less features which were the chin and mouth. Similarly, a Category 3 model was constructed for this particular facial image. Once these models were constructed, the images were split into a training set and a testing set in a ratio of 0.7:0.3. The testing images were then used to obtain accuracy and error metrics. Five-fold cross validation was utilised so as to use all images in the entire set for testing. The average accuracy and error metrics from the five-fold cross validation was used as the rate for that particular metric.

6.2 CONCLUSIONS

Experiments that were carried out on the three databases that were used for bench marking uncovered a surprising conclusion. When analysing which specific feature or sub-region of the face offered the best discrimination, it was found out that the forehead had the best discrimination at an effective 87% recognition rate. This rate was the effective average recognition rate that the 16 sub-models that had the forehead feature had the best average discrimination. Unsurprisingly the eye feature and its 16 models, came in second at an average of 85%. This was followed by the nose and chin features and their 16 models had an average of 80%. The mouth feature had the least discrimination at 77%. The forehead had the best discrimination. This can be attributed to the fact that the forehead region contained the brows as well as parts of hair. Since people have brows that are shaped differently, this partially contributed to the differences between individuals. Such high discrimination for this feature can also be attributed to the fact that the subjects had different hair styles as well as varying amounts of hair. Some subjects had full and thick hair, some subjects were bald, some were balding, some subjects had wigs with fringes, some subjects had frisky hair and some had relaxed hair. These major variations in hair styles contributed to the forehead feature, and its other 15 models, having the best discrimination rate. Having such a high discrimination rate means that even if a person has

partial occlusion on the mouth region, due to a scarf, the F feature and its other models can be used for recognition resulting in a high enough recognition rate. The mouth was found to have the least average discrimination. One reason could be that the mouths of most subjects had a similar shape, size and thickness. The three databases contained predominantly Caucasian subjects so this also played a role in the mouths having a similar shape and thickness. The mouth region does not contain that many variations from subject to subject, thus the proposed technique failed to properly discriminate between the subjects owing to the close similarities.

The other major focus of research presented in this dissertation was to find out which combination of regions on the face could potentially match or even rival other methods that utilise the entire face. Category 3 features only used a combination of three out of five features, this experimentally simulated the situation where two of the five features were occluded. These two features basically did not contribute to recognition there simulating partial occlusion on the face. For example, a Category 3 model used only the forehead, eyes and nose combination, therefore it was termed as EFN model. This translated to the mouth and chin being occluded. Category 4 features used a combination of four of the five features for recognition. Similarly, this also meant that one out of the five features also did not contribute to recognition, thus simulating partial occlusion. In this case a CEMN model used four features except for the forehead. In comparison with other methods in literature, when it comes to the recognition rate, the proposed technique outperformed various other methods.

When comparing against the AR database, Category 3 models at 86%, Category 4 models at 90% and a Category 5 model at 91% recognition rates outperformed the following methods taken straight from literature; KPCA at 83% [140], LGBP at 80% [141], PSVM at 82% [142], PUM at 80% [143], SRC at 84% [144] and PUDMNN at 82% [145]. When considering the ORL database, Category 4 models at 87% and a Category 5 model at 89% outperformed the following methods taken from literature; PUDMNN at 84% [145], Gabor-SVD at 75% [148], WPCA at 86% [149] and 2DPCA-ICA at 83% [150]. The same trend continued on the Yale Faces database as well, Category 2 models at 79%, Category 3 models at 83%, Category 4 models at 85% and a Category 5 model at 87% outperformed other methods from literature. These methods included the 2DPCA-ICA at 79% [150], ALBP at 62% [151], DLBP at 56% [151], LBP at 61% [151], and finally LDP at 71% [151].

What is undeniable and can be drawn from these comparisons is that the proposed technique and using Category 3, 4 and 5 models outperforms several methods from literature. Even though the proposed technique utilises less information from the face, since partial occlusions were simulated onto the image, the proposed technique still outshone various other methods from literature. This makes the proposed technique an effective and robust facial recognition system when it comes to images with

partial occlusions when comparing with the other methods. The proposed technique's ability to do more with less information is invaluable in situations where parts of the face are inadvertently or purposefully occluded.

6.3 FUTURE WORK

The proposed technique used weighing and image segmentation to achieve high recognition rates and low FAR and FRR. However, this in no way suggests it can not be improved or modified or extended. Some modifications that could be done are to include the cheeks and the ears as two other independent features that could be used to build a seven-tier recognition system which will inevitably lead to 127 unique combinations of seven features.

The weighing technique can be extended further as well, by adding more centres other than the five proposed in this dissertation. These centres would be able to closely match all the regions of the face. Tuning the parameters for the weighing function to suit the entire database and not use unique parameters for each individual, as proposed in this dissertation, can be another place that can be investigated further. This makes the recognition process more time efficient, as only one set of parameters would be used for a single subject's class of images.

REFERENCES

[1] K. Dharavath, G. Amarnath, F. A. Talukdar, and R. H. Laskar, "Impact of image preprocessing on face recognition: A comparative analysis," *2014 International Conference on Communication and Signal Processing*, pp. 631–635, 2014.

[2] M. Hu, Q. Zhang, and Z. Wang, "Application of rough sets to image pre-processing for face detection," *2008 International Conference on Information and Automation*, pp. 545–548, 2008.

[3] S. R. Benedict and J. S. Kumar, "Geometric shaped facial feature extraction for face recognition," *2016 IEEE International Conference on Advances in Computer Applications*, pp. 275–278, 2016.

[4] A. V. Nefian and M. H. Hayes, "Hidden Markov models for face recognition," *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 5, pp. 2721–2724, 1998.

[5] S. Singh and S. Prasad, "Techniques and challenges of face recognition: A critical review," *Procedia Computer Science*, vol. 143, pp. 536–543, 2018.

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

[7] F. Tarres and A. Rama, "A novel method for face recognition under partial occlusion or facial expression variations," *47th International Symposium Electronics in Marine, 2005.*, pp. 163–166, 2005.

[8] W. Zhang, S. Shan, X. Chen, and W. Gao, "Local Gabor binary patterns based on Kullback–Leibler divergence for partially occluded face recognition," *IEEE Signal Processing Letters*, vol. 14, pp. 875–878, 2007.

[9] F. Tsalakanidou, S. Malassiotis, and M. G. Strintzis, "Face localization and authentication using

color and depth images," *IEEE Transactions on Image Processing*, vol. 14, pp. 152–168, 2005.

[10] Y. Long, F. Zhu, L. Shao, and J. Han, "Face recognition with a small occluded training set using spatial and statistical pooling," *Information Sciences*, vol. 430, pp. 634–644, 2018.

[11] Y. Fu, X. Wu, Y. Wen, and Y. Xiang, "Efficient locality-constrained occlusion coding for face recognition," *Neurocomputing*, vol. 260, pp. 104–111, 2017.

[12] H. J. Oh, K. M. Lee, and S. U. Lee, "Occlusion invariant face recognition using selective local non-negative matrix factorization basis images," *Image and Vision Computing*, vol. 26, pp. 1515–1523, 2008.

[13] S. Lin, B. Liu, and J. Lin, "Combining speeded-up robust features with principal component analysis in face recognition system," *International Journal of Innovative Computing, Information and Control*, vol. 8, pp. 8545–8558, Dec 2012.

[14] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," *Proceedings of the 9th European conference on Computer Vision*, vol. 3951, pp. 404–417, Jul 2006.

[15] S.-M. Huang and J.-F. Yang, "Subface hidden Markov models coupled with a universal occlusion model for partially occluded face recognition," *The Institute of Engineering and Technology on Biometrics*, vol. 1, pp. 149–159, 2012.

[16] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Face recognition using Laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 328–340, 2005.

[17] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 210–227, 2009.

[18] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, vol. 35, pp. 399–458, 2003.

[19] B. Xiao-man and R. Qiu-qi, "An improved wpca plus lda," *2006 8th international Conference on Signal Processing*, vol. 2, pp. –, 2006.

[20] T. A. Davis, "Direct methods for sparse linear systems," in *Fundamentals of algorithms*, 2006.

[21] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: a convolutional neural-network approach," *IEEE transactions on neural networks*, vol. 81, pp. 98–113, 1997.

[22] J.-T. Chien and C.-C. Wu, "Discriminant waveletfaces and nearest feature classifiers for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1644–1649, 2002.

[23] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 2106–2112, 2010.

[24] Z.-Q. Hong, "Algebraic feature extraction of image for recognition," *Pattern Recognition*, vol. 24, pp. 211–219, 1991.

[25] B. Manjunath, R. Chellappa, and C. Malsburg, "A feature based approach to face recognition," *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 373–378, 1992.

[26] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 2037–2041, 2006.

[27] Z. Xu and H. R. Wu, "Shape feature based extraction for face recognition," *2009 4th IEEE Conference on Industrial Electronics and Applications*, pp. 3043–3048, 2009.

[28] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 586–591, June 1991.

[29] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 2037–2041, 2006.

[30] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 711–720, 1997.

[31] M. Hanmandlu, D. Gupta, and S. Vasikarla, "Face recognition using elastic bunch graph matching," *2013 IEEE Applied Imagery Pattern Recognition Workshop*, pp. 1–7, 2013.

[32] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: a review of classification and combining techniques," *Artificial Intelligence Review*, vol. 26, pp. 159–190, 2006.

[33] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, pp. 131–156,

1997.

[34] M. Sidana. Types of classification algorithms in Machine Learning. [Accessed 18 June 2019]. [Online]. Available: https: //medium.com/@Mandysidana/machine-learning-types-of-classification-9497bd4f2e14

[35] S. An, W. Liu, and S. Venkatesh, "Face recognition using kernel ridge regression," *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–7, 2007.

[36] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 2004.

[37] W. W. Bledsoe, "Man-machine facial recognition," *Panoramic Research Institute*, Technical Report PRI 22, August 1966.

[38] A. J. Goldstein, L. D. Harmon, and A. B. Lesk, "Identification of human faces," *Proceedings of the IEEE*, vol. 59, no. 5, pp. 748–760, May 1971.

[39] S. Kaushik, R. B. Dubey, and A. Madan, "Study of face recognition techniques," *International Journal of Advanced Computer Research*, vol. 4, no. 4, December 2014.

[40] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Journal of the Optical Society of America*, vol. 4, no. 3, p. 519, March 1987.

[41] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, July 1997.

[42] F. S. Samaria and F. Fallside, "Face identification and feature extraction using hidden Markov models," in *Image Processang: Theory and Applications*, 1993.

[43] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, Dec 1994, pp. 138–142.

[44] F. Samaria, "Face segmentation for identification using hidden Markov models." in *British Machine Vision Conference*, 1993, pp. 1–10.

[45] M. Pietikainen, A. Hadid, G. Zhao, and T. Ahonen, "Computer vision using local binary patterns," *Computational Imaging and Vision Computer Vision Using Local Binary Patterns*, 2004.

[46] C. Ding and D. Tao, "Trunk-branch ensemble convolutional neural networks for video-based face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 1002–1014, April 2018.

[47] C. Ding and D. Tao, "Trunk-Branch Ensemble Convolutional Neural Networks for video-based face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 1002–1014, April 2018.

[48] C. Brooke and D. Tao, "Recommendation ITU-R BT.709-6: Parameter values for the HDTV standards for production and international programme exchange," *International Telecommunication Union*, vol. 40, no. 4, June 2015.

[49] A. D. Whitehead, "Fast feature-based video segmentation and annotation," *7th International Symposium on Signal Processing and Its Applications*, pp. 1–44, Jan. 2003.

[50] M. Kaur and K. Verma, "A novel hybrid technique for low exposure image enhancement using sub-image histogram equilization and artificial neural network," in *2016 International Conference on Inventive Computation Technologies*, vol. 2, Aug 2016, pp. 1–5.

[51] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, July 1997.

[52] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data - with application to face recognition," *Pattern Recognition*, vol. 34, pp. 2067–2070, 2001.

[53] H. Gao and J. W. Davis, "Why direct LDA is not equivalent to LDA," *Pattern Recognition*, vol. 39, pp. 1002–1006, 2006.

[54] J. R. Price and T. F. Gee, "Face recognition using direct, weighted linear discriminant analysis and modular subspaces," *Pattern Recognition*, vol. 38, pp. 209–219, 2005.

[55] B. Xiaoman and R. Qiuqi, "An improved WPCA plus LDA," in *2006 8th international Conference on Signal Processing*, vol. 2, Feb 2006.

[56] R. Brunelli and T. A. Poggio, "Face recognition: Features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1042–1052, 1993.

[57] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," *1994 Proceedings of IEEE Conference on Computer Vision and Pattern*

Recognition, pp. 84–91, 1994.

[58] G. Pushpalatha, H. Jyothi, S. Nithya, and C. Shakuntala, "A review of discriminant analysis algorithms for face recognition using similarity measures," in *2017 International Conference on Recent Advances in Electronics and Communication Technology*, March 2017, pp. 249–253.

[59] Q. Jin, Y. Huang, and C. Wang, "Modular discriminant analysis and its applications," *Artificial Intelligence Review*, vol. 39, pp. 285–303, 2011.

[60] A. Nefian and M. Hayes, "Hidden markov models for face recognition," *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 5, pp. 2721–2724, 1998.

[61] V. Kohir and B. Desai, "Face recognition using a DCT-HMM approach." *Proceedings Fourth IEEE Workshop on Applications of Computer Vision*, October 1998.

[62] J. Bobulski, "2DHMM-based face recognition method," *Image Processing and Communications Challenges 7. Advances in Intelligent Systems and Computing*, vol. 389, 2015.

[63] V. Bevilacqua, L. Cariello, G. Carro, D. Daleno, and G. Mastronardi, "A face recognition system based on pseudo 2D HMM applied to neural network coefficients," *Soft Computing*, vol. 12, pp. 615–621, 2008.

[64] Z. M. Hafed and M. D. Levine, "Face recognition using the discrete cosine transform," *International Journal of Computer Vision*, vol. 43, pp. 167–188, 2001.

[65] L. L. Thomas, C. Gopakumar, and A. Thomas, "Face recognition based on Gabor wavelet and backpropagation neural network," *Multidisciplinary Digital Publishing Institute*, 2016.

[66] B. Kepenekci, F. Boray Tek, and G. Bozdagi Akar, "Occluded face recognition based on Gabor wavelets," in *Proceedings. International Conference on Image Processing*, vol. 1, Sep. 2002, pp. I–I.

[67] D. N. Pritha, L. Savitha, and S. S. Shylaja, "Face recognition by feed-forward neural network using Laplacian of Gaussian filter and singular value decomposition," in *2010 First International Conference on Integrated Intelligent Computing*, Aug 2010, pp. 56–61.

[68] V. P. Vishwakarma, "Deterministic learning machine for face recognition with multi-model feature extraction," in *2016 Ninth International Conference on Contemporary Computing*, Aug 2016, pp. 1–6.

[69] S. Prasad, S. S. Panda, G. S. Deepthi, and V. Anisha, "A face recognition using PCA and feed forward neural networks," *IEEE International Joint Conference on Neural Networks*, 2011.

[70] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1701–1708.

[71] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *2015 IEEE Conference on Computer Vision and Pattern Recognition*, June 2015, pp. 2892–2900.

[72] C. Xiong, X. Zhao, D. Tang, K. Jayashree, S. Yan, and T. Kim, "Conditional convolutional neural network for modality-aware face recognition," in *2015 IEEE International Conference on Computer Vision*, Dec 2015, pp. 3667–3675.

[73] L. Tian, C. Fan, Y. Ming, and Y. Jin, "Stacked PCA Network (SPCANet): An effective deep learning for face recognition," in *2015 IEEE International Conference on Digital Signal Processing*, July 2015, pp. 1039–1043.

[74] W. Cao, L. Li, and X. Lv, "Kernel function characteristic analysis based on support vector machine in face recognition," in *2007 International Conference on Machine Learning and Cybernetics*, vol. 5, no. 7, Aug 2007, pp. 2869–2873.

[75] E. Osuna and R. F. F. Girosi, "Training support vector machine: an application to face detection," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 130–136, 1997.

[76] P. J. Philips, M. I. Jordan, J. Kearns, and S. A. Solla, "Support vector machines applied to face recognition," *Advances in Neural Information Processing Systems*, vol. 11, pp. 803–809, 1998.

[77] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[78] S. R. Safavian and D. A. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, pp. 660–674, 1991.

[79] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[80] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: A classification and regression tool for compound classification and QSAR modeling," *Journal of chemical information and computer sciences*, vol. 43, no. 6, pp. 1947–1958, 2003.

[81] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes," in *Neural Information Processing Systems*, 2001.

[82] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Computers Security*, vol. 21, pp. 439–448, 2002.

[83] A. R. Chadha, P. P. Vaidya, and M. M. Roja, "Face recognition using discrete cosine transform for global and local features," in *International Conference on recent advancements in Electrical, Electronic and Control Engineering*, Dec 2011, pp. 502 – 505.

[84] Y. Hongtao, Q. Jiaqing, and F. Ping, "Face recognition with discrete cosine transform," in *2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control*, Dec 2012, pp. 802–805.

[85] Z. M. Hafed and M. D. Levine, "Face recognition using the discrete cosine transform," *International Journal of Computer Vision*, vol. 43, pp. 167–188, 2001.

[86] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, pp. 90–93, 1974.

[87] A. M. Martinez and R. Benavente, "The AR Face Database," *CVT Technical Report 24*, June 1998.

[88] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*, August 1994, pp. 138-142.

[89] "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, July 1997, pp. 711-720.

[90] R. A. Hummel, "Histogram modification techniques," *Computer Graphics and Image Processing*, vol. 4, no. 3, pp. 209–224, 1975.

[91] S. Khayam. (2003) The discrete cosine transform (DCT): Theory and application. ECE 802 – 602: Information Theory and Coding. [Online]. Available: http://cs.uccs.edu/ cs525/video/dctKhayam.pdf

[92] L. Wang, Y. Zhang, and J. Feng, "On the euclidean distance of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1334–1339, 2005.

[93] F. Nie, H. Huang, X. Cai, and C. H. Q. Ding, "Efficient and robust feature selection via joint $\ell 2$, $\ell 1$-norms minimization," in *Neural Information Processing Systems*, 2010.

[94] A. Hyvärinen, "Survey on independent component analysis," 1999.

[95] P. Comon, "Independent component analysis," 1992.

[96] P. Common, "Independent component analysis, a new concept," *Signal Processing*, vol. 36, pp. 287-314, 1994.

[97] M. S. Bartlett, M. Lades, and T. J. Sejnowski, "Independent component representations for face recognition," in *Human Vision and Electronic Imaging*, 1998.

[98] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face recognition by independent component analysis," *IEEE transactions on neural networks*, vol. 136, pp. 1450-64, 2002.

[99] C. Liu and H. Wechsler, "Independent component analysis of Gabor features for face recognition," *IEEE transactions on neural networks*, vol. 144, pp. 919-28, 2003.

[100] J. F. Cardoso, "Multidimensional independent component analysis," *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4, pp. 1941-1944, 1998.

[101] G. Sahonero-Alvarez, "A comparison of SOBI, FastICA, JADE and Infomax algorithms," *Proceedings of The 8th International Multi-Conference on Complexity, Informatics and Cybernetics*, pp. 17–22, Mar 2017.

[102] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks : The official journal of the International Neural Network Society*, vol. 134, no. 5, pp. 411-430, 2000.

[103] E. Oja and Z. Yuan, "The FastICA algorithm revisited: Convergence analysis," *IEEE Transactions on Neural Networks*, vol. 17, pp. 1370-1381, 2006.

[104] J. Cardoso, "Infomax and maximum likelihood for blind source separation," *IEEE Signal Processing Letters*, vol. 4, pp. 112-114, 1997.

[105] Yi-qiong Xu, Bi-cheng Li, and Bo Wang, "Face recognition by fast independent component analysis and genetic algorithm," in *The Fourth International Conference on Computer and Information Technology*, Sep. 2004, pp. 194-198.

[106] Y. Zhang and J. Guo, "Weighted Fisher non-negative matrix factorization for face recognition," in *2009 Second International Symposium on Knowledge Acquisition and Modeling*, vol. 1, Nov 2009, pp. 232-235.

[107] F. Purnomo, D. Suhartono, M. Shodiq, A. Susanto, S. Raharja, and R. W. Kurniawan, "Face recognition using Gabor wavelet and non-negative matrix factorization," in *2015 SAI Intelligent Systems Conference*, Nov 2015, pp. 788-792.

[108] T. Virtanen, A. T. Cemgil, and S. J. Godsill, "Bayesian extensions to non-negative matrix factorisation for audio signal modelling," *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1825-1828, 2008.

[109] W. P. Koppen, W. J. Christmas, D. J. M. Crouch, W. F. Bodmer, and J. Kittler, "Extending non-negative matrix factorisation to 3d registered data," *2016 International Conference on Biometrics*, pp. 1-8, 2016.

[110] X. Long, H. Lu, and Y. Peng, "Sparse non-negative matrix factorization based on spatial pyramid matching for face recognition," in *2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 1, Aug 2013, pp. 82-85.

[111] P. D. O'Grady and B. A. Pearlmutter, "Convolutive non-negative matrix factorisation with a sparseness constraint," *2006 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, pp. 427-432, 2006.

[112] W. Chen, Q. Wang, B. Pan, and Y. Li, "Local and non-local feature-based kernel non-negative matrix factorization method for face recognition," in *2016 12th International Conference on Computational Intelligence and Security*, Dec 2016, pp. 107-110.

[113] N. Gillis, "The why and how of nonnegative matrix factorization," *Machine Learning and Pattern Recognition Series*, vol. 1401.5226, pp. 257-291, 2014.

[114] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788-791, 1999.

[115] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local Gabor binary pattern histogram sequence: a novel non-statistical model for face representation and recognition," *Tenth IEEE International Conference on Computer Vision*, vol. 1, pp. 786-791, 2005.

[116] T. S. Lee, "Image representation using 2D Gabor wavelets," *IEEE Transactions on Pattern*

Analysis and Machine Intelligence, vol. 18, pp. 959-971, 1996.

[117] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition," *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 114, pp. 467-476, 2002.

[118] I. S. Fogel and D. Sagi, "Gabor filters as texture discriminator," *Biological Cybernetics*, vol. 61, pp. 103-113, 1989.

[119] C. Liu and H. Wechsler, "Independent component analysis of Gabor features for face recognition," *IEEE transactions on neural networks*, vol. 144, pp. 919-28, 2003.

[120] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pattern Recognition*, vol. 24, pp. 1167-1186, 1991.

[121] X. Peng, C. Lu, Z. Yi, and H. Tang, "Connections between nuclear-norm and Frobenius-norm-based representations," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, pp. 218-224, 2018.

[122] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 210–227, 2009.

[123] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition," *2011 International Conference on Computer Vision*, pp. 471–478, 2011.

[124] J. Lai and X. Jiang, "Modular weighted global sparse representation for robust face recognition," *IEEE Signal Processing Letters*, vol. 19, no. 9, pp. 571–574, Sep. 2012.

[125] C. Ma, P. Xu, and M. Shang, "Face recognition via gradient projection for sparse representation," in *2013 6th International Congress on Image and Signal Processing*, vol. 2, Dec 2013, pp. 763–767.

[126] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," *2011 International Conference on Computer Vision*, pp. 543–550, 2011.

[127] Daming Zhang, Xueyong Zhang, Huayong Liu, and Lu Li, "Image synthesis for sparse representation based multi-pose face recognition," in *2014 IEEE Workshop on Advanced Research and Technology in Industry Applications*, Sep. 2014, pp. 1051–1054.

[128] J. Zeng, Y. Zhai, and J. Gan, "The sparse representation and smoothed ℓ-0 algorithm for face recognition," in *2015 International Conference on Wavelet Analysis and Pattern Recognition*, July 2015, pp. 34–38.

[129] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[130] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Neural Information Processing Systems*, 2006.

[131] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*, vol. 7700, 2012, pp. 421-436.

[132] P. Blache, H. Rabah, and A. Amira, "High level prototyping and fpga implementation of the orthogonal matching pursuit algorithm," in *2012 11th International Conference on Information Science, Signal Processing and their Applications*, July 2012, pp. 1336–1340.

[133] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pp. 40–44 vol.1, 1993.

[134] S. N. Albarqouni, "Introduction to sparse methods," May 2015. [Online]. Available: https://www.slideshare.net/sbaraqouni/shadi-sparsity

[135] J. F. Pereira, G. D. C. Cavalcanti, and T. I. Ren, "Modular image principal component analysis for face recognition," in *2009 International Joint Conference on Neural Networks*, vol. 40, no. 12, 2009, pp. 2481–2486.

[136] Jian Yang, D. Zhang, A. F. Frangi, and Jing-yu Yang, "Two-dimensional pca: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, 2004.

[137] L. E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," in *Inequalities III: Proceedings of the Third Symposium on Inequalities*. Academic Press, 1972, pp. 1–8.

[138] M. Visani, C. Garcia, and J.-M. Jolion, "Face recognition using Modular Bilinear Discriminant Analysis," in *Visual Information and Information Systems*, S. Bres and R. Laurini, Eds., 2006, pp. 24–34.

[139] S. Moore and R. Bowden, "Local binary patterns for multi-view facial expression recognition," *Computer Vision and Image Understanding*, vol. 115, pp. 541–558, 2011.

[140] S. Gundimada and V. K. Asari, "Facial recognition using multisensor images based on localized kernel eigen spaces," *IEEE Transactions on Image Processing*, vol. 18, no. 6, pp. 1314–1325, 2009.

[141] W. Zhang, S. Shan, X. Chen, and W. Gao, "Local gabor binary patterns based on kullback–leibler divergence for partially occluded face recognition," *IEEE Signal Processing Letters*, vol. 14, no. 11, pp. 875–878, 2007.

[142] G. Guo, S. Li, and K. Chan, "Face recognition by support vector machines," *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 196–201, 2000.

[143] J. Lin, J. Ming, and D. Crookes, "Robust face recognition with partially occluded images based on a single or a small number of training samples," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 881–884.

[144] J. Wright, A. Ganesh, Zihan Zhou, A. Wagner, and Yi Ma, "Demo: Robust face recognition via sparse representation," in *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*, 2008, pp. 1–2.

[145] J. Lin, J. Ming, and D. Crookes, "A probabilistic union approach to robust face recognition with partial distortion and occlusion," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2008, pp. 993–996.

[146] K. Xu, X. Wang, Z. Hu, and Z. Zhang, "3d face recognition based on twin neural network combining deep map and texture," in *2019 IEEE 19th International Conference on Communication Technology*, 2019, pp. 1665–1668.

[147] C. Xie, L. Li, H. Wang, and J. He, "Face recognition based hybrid fuzzy hidden Markov models," in *2010 2nd International Conference on Signal Processing Systems*, vol. 2, 2010, pp. V2–661–V2–665.

[148] Lim Song Li and N. Yahya, "Face recognition technique using Gabor wavelets and singular value decomposition (SVD)," in *2014 IEEE International Conference on Control System, Computing and Engineering*, 2014, pp. 455–459.

[149] W. Puyati and A. Walairacht, "Efficiency improvement for unconstrained face recognition by weightening probability values of modular PCA and wavelet PCA," in *2008 10th International Conference on Advanced Communication Technology*, vol. 2, Feb 2008, pp. 1449–1453.

[150] Xingfu Zhang and Xiangmin Ren, "Two-dimensional principal component analysis based independent component analysis for face recognition," in *2011 International Conference on Multimedia Technology*, 2011, pp. 934–936.

[151] K. Meena and A. Suruliandi, "Performance evaluation of local binary patterns and its derivatives for face recognition," in *2011 International Conference on Emerging Trends in Electrical and Computer Technology*, 2011, pp. 742–746.

[152] Shang-Hung Lin, Sun-Yuan Kung, and Long-Ji Lin, "Face recognition/detection by probabilistic decision-based neural network," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 114–132, 1997.

[153] J. Ravi, S. S. Tevaramani, and K. B. Raja, "Face recognition using DT-CWT and LBP features," in *2012 International Conference on Computing, Communication and Applications*, 2012, pp. 1–6.

[154] I. G. P. S. Wijaya, A. Y. Husodo, and A. H. Jatmika, "Real time face recognition engine using compact features for electronics key," in *2016 International Seminar on Intelligent Technology and Its Applications*, July 2016, pp. 151–156.

[155] G. Prabhu Teja and S. Ravi, "Face recognition using subspaces techniques," in *2012 International Conference on Recent Trends in Information Technology*, 2012, pp. 103–107.

[156] R. Kaur, D. Sharma, and A. Verma, "An advance 2D face recognition by feature extraction (ICA) and optimize multilayer architecture," in *2017 4th International Conference on Signal Processing, Computing and Control*, 2017, pp. 122–129.

[157] R. Toufiq and M. R. Islam, "Face recognition system using PCA-ANN technique with feature fusion method," in *2014 International Conference on Electrical Engineering and Information Communication Technology*, 2014, pp. 1–5.

[158] S. Sardar, G. Tewari, and K. A. Babu, "A hardware/software co-design model for face recognition using cognimem neural network chip," in *2011 International Conference on Image Information Processing*, 2011, pp. 1–6.