

**FACE RECOGNITION IN CRIMINAL
INVESTIGATIONS
(UNCONTROLLED ENVIROMENT)**

LIEW YAO QIN

UNIVERSITI TUNKU ABDUL RAHMAN

**FACE RECOGNITION IN CRIMINAL INVESTIGATIONS
(UNCONTROLLED ENVIRONMENT)**

LIEW YAO QIN

**A project report submitted in partial fulfilment of the
requirements for the award of Bachelor of Engineering
(Hons.) Electrical and Electronic Engineering**

**Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman**

April 2017

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : _____

Name : LIEW YAO QIN

ID No. : 14UEB00275

Date : AUGUST 28 2017

APPROVAL FOR SUBMISSION

I certify that this project report entitled “**FACE RECOGNITION IN CRIMINAL INVESTIGATIONS (UNCONTROLLED ENVIRONMENT)**” was prepared by **LIEW YAO QIN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons.) Electrical and Electronic Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : _____

Supervisor : _____

Date : _____

The copyright of this report belongs to the author under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of Universiti Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2017, Liew Yao Qin. All right reserved.

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Mr See Yuen Chark for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to my loving parents and friends who had helped and given me encouragement. I am also grateful to UTAR for providing me adequate resources in implementing my project.

ABSTRACT

The application of face recognition through public surveillance cameras in criminal investigations is non-intrusive, inconspicuous and faster compared with fingerprint or DNA-sampling biometric evaluation methods. It assists the law enforcement agency during the preliminary investigation to narrow down the candidate list of suspects or person-of-interest. In this project, the fusion between Gabor filters and Maximum Response (MR) filters with Random Forests classifier is proposed for face recognition in criminal investigation. The proposed method differs from the algorithms such as deep neural networks in which the deep neural networks require larger training datasets. However, it is difficult to obtain large volume of the face images of the same individuals in reality. Deep neural networks are also more computation exhausting. The Gabor and MR filters are the facial features extractor. The Gabor filters are the hybrid of the Gabor magnitude filters and oriented Gabor phase congruency (OGPC) filters. Gabor magnitude filters produce the magnitude response while the OGPC filters produce the phase response of Gabor filters. The MR filters produce the edge- and bar-anisotropic filter responses and isotropic filter responses. The variable selection using Monte Carlo Uninformative Variable Elimination Partial Least Squares Regression (MC-UVE-PLSR) is used to pick the most useful features in order to save computation costs and time which is crucial in criminal investigations. Random Forests is used in the classification of the generated feature vectors. The algorithm is applied in uncontrolled environment where the unconstrained parameters of facial images such as uncontrolled illumination, pose and expression variations commonly present in the tape recorded by surveillance cameras during criminal investigation. The algorithm performance is evaluated using two unconstrained facial image databases: Labelled Faces in the Wild and Unconstrained Facial Images (UFI). The images of the databases include different illumination, face expressions and pose variations. The implemented method achieved 81.28% and 67.33% of recognition rates and 97.07% and 93.06% of Receiver Operating Characteristics (ROC) curve on LFW and UFI databases.

TABLE OF CONTENTS

DECLARATION	iii
APPROVAL FOR SUBMISSION	iv
ACKNOWLEDGEMENTS	vi
ABSTRACT	vii
TABLE OF CONTENTS	viii
LIST OF TABLES	xii
LIST OF FIGURES	xiii
LIST OF SYMBOLS / ABBREVIATIONS	xvi
LIST OF APPENDICES	xvii

CHAPTER

1	INTRODUCTION	1
	1.1 General Introduction	1
	1.2 Importance of the Study	1
	1.3 Problem Statement	2
	1.4 Aims and Objectives	3
	1.5 Scope and Limitation of the Study	4
	1.6 Contribution of the Study	4

1.7	Outline of the Report	4
2	LITERATURE REVIEW	6
2.1	Introduction	6
2.1.1	Face Recognition System	6
2.1.2	Face Detection Method	10
2.1.3	Feature Extraction	12
2.1.4	Feature Selection	19
2.2	Face Recognition	21
2.2.1	Subspace-based Method	21
2.2.2	Learning-based Method	25
2.3	Summary	29
3	METHODOLOGY AND WORK PLAN	30
3.1	Introduction	30
3.2	Databases	32
3.2.1	Labelled Faces in the Wild (LFW)	32
3.2.2	Unconstrained Facial Images (UFI)	34
3.3	Image Pre-Processing	35
3.4	Feature Extraction	36
3.4.1	Gabor Filters	36
3.4.2	Oriented Gabor Phase Congruency Filters	39
3.4.3	Maximum Response Filters	40
3.5	Feature Selection	42
3.5.1	Monte Carlo Uninformative Variable Elimination Partial Least Squares Regression (MC-UVE-PLSR)	42

3.6	Classification and Prediction Using Learning Framework	43
3.6.1	Random Forests	44
3.7	Fusion of Gabor Filters and Maximum Response Filters	47
3.8	Software	49
4	RESULTS AND DISCUSSIONS	50
4.1	Introduction	50
4.2	The Pre-Processing Phase	50
4.2.1	Face Detection	50
4.2.2	Conversion into Grayscaled Format	51
4.2.3	Tan-Triggs Illumination Normalization	51
4.3	Feature Extraction	53
4.3.1	Gabor Filters	53
4.3.2	Maximum Response (MR) Filters	57
4.4	Feature Selection	58
4.5	Classification using Random Forests	61
4.6	Fusion of the Random Forest Prediction Scores of Gabor Magnitude, OGPC and MR filters	64
4.7	Finalization of Parameters	65
4.8	Comparison with Other Existing Algorithms	68
4.8.1	The Use of Cross Validation in the Evaluation of Algorithms	68
4.8.2	Labelled Faces in the Wild	70
4.8.3	Unconstrained Facial Images (UFI)	78
4.8.4	Summary of the Comparison of Algorithms using LFW and UFI Databases	85

5	CONCLUSIONS AND RECOMMENDATIONS	86
5.1	Conclusions	86
5.2	Challenges and Recommendations for future work	86
	REFERENCES	88

LIST OF TABLES

Table 3.1:	Experimental Setup of LFW Database	32
Table 3.2:	Experimental Setup of UFI Database	34
Table 4.1:	Study of the effect of Tan-Triggs illumination normalization	52
Table 4.2:	Study of the effect of tuning the Gabor filter parameter on the recognition rate	55
Table 4.3:	Study of the effect number of features and feature selection on the recognition rates	58
Table 4.4 :	Summarized parameter settings of LFW database	66
Table 4.5:	Summarized parameter settings of UFI database	67
Table 4.6:	Comparison of Classification Accuracy among Different Algorithms on	73
Table 4.7:	Comparison of ROC AUC among algorithms on LFW database using the	75
Table 4.8:	Comparison of Classification Accuracy among Algorithms on UFI Database (Ufi.kiv.zcu.cz., 2017)	82
Table 4.9:	Comparison of ROC AUC between Algorithms	84

LIST OF FIGURES

Figure 2.1: The Phases of the Face Recognition System	6
Figure 2.2: Feature Extraction Stage	9
Figure 2.3: Five Haar-like Patterns (Datta, Datta and Banerjee, 2015)	10
Figure 2.4: Integral image	11
Figure 2.5: Two-rectangle feature from integral image	11
Figure 2.6: Visualization of LBP	12
Figure 2.7: Circular LBP	13
Figure 2.8: Forty Gabor filters with five scales and eight orientations	15
Figure 2.9: Gabor Representation of Facial Image (Zhou, Yin, and Zhang, 2011)	16
Figure 2.10: Maximum Filter Response (Caenen, 2004)	19
Figure 2.11: : Geometric Feature-based Method	24
Figure 2.12: Support Vector Machine (Docs.opencv.org, 2017)	26
Figure 2.13: K-Nearest Neighbours (Srivastava, 2017)	27
Figure 3.1: Flow diagram of training process	30
Figure 3.2: Flow diagram of testing process	31
Figure 3.3: Examples of face images in LFW database	33
Figure 3.4: Examples of face images in UFI database	35
Figure 3.5: Image Pre-processing	36
Figure 3.6: The filter output of the MR filters. First three rows are edge filters. The fourth until sixth rows are bar filters. The remaining two filters are isotropic Gaussian and Laplacian of Gaussian filters	42

Figure 3.7:	Flowchart of the random forest (Rodriguez-Galiano et al., 2015)	44
Figure 3.8:	Visualization of decision split	47
Figure 3.9:	Fusion between Gabor and MR filters	49
Figure 4.1:	The output of the face detection after resizing	50
Figure 4.2:	Conversion into grayscaled image	51
Figure 4.3:	Effect of Illumination normalization to counter the influence of shading or overexposure of light	52
Figure 4.4:	Gabor magnitude response	54
Figure 4.5:	Gabor Phase Response	54
Figure 4.6:	Even symmetric filter transfer function	56
Figure 4.7:	MR Filter Response	57
Figure 4.8 :	The positive-valued OOB Feature Importance among the pools of 4000 features	60
Figure 4.9:	The positive-valued OOB Feature Importance among the pools of 2000 features	61
Figure 4.10:	Partial visualization of the decision tree with class label at the nodes	62
Figure 4.11:	Effect of number of decision trees on Recognition rate	63
Figure 4.12:	Effect of number of grown decision trees on Out-of-bag mean square error	63
Figure 4.13:	Effect of Gabor phase to Gabor magnitude ratio on Recognition rate	65
Figure 4.14:	The graph of Recognition rate against the MR-Gabor ratio	65
Figure 4.15:	Cross validation	69
Figure 4.16:	Confusion Matrix of Gabor-MR-fusion with Random Forest on LFW database	71
Figure 4.17:	Confusion Matrix of Gabor-MR-fusion with Random Forest on LFW database (continued)	72

Figure 4.18: Comparison of Accuracy Rate among Algorithms on LFW Database	74
Figure 4.19: The ROC distribution curve	75
Figure 4.20: Comparison of ROC AUC among algorithms on LFW database	76
Figure 4.21: Visualization of ROC Curve of existing algorithms (only ROC of some algorithms are shown here)	77
Figure 4.22: Detection Error Tradeoff Curve on LFW database	78
Figure 4.23: Confusion Matrix of Gabor-MR-fusion with Random Forest on UFI database	79
Figure 4.24: Confusion Matrix of Gabor-MR-fusion with Random Forest on UFI database (continued)	80
Figure 4.25: Confusion Matrix of Gabor-MR-fusion with Random Forest on UFI database (continued)	81
Figure 4.26: Comparison of Classification Accuracy among Algorithms on UFI Database	83
Figure 4.27: Visualization ROC Curve of Gabor-MR-fusion Random Forest on UFI Database	84
Figure 4.28: Visualization DET Curve of Gabor-MR-fusion Random Forest on UFI Database	85

LIST OF SYMBOLS / ABBREVIATIONS

CGC	Complete Gabor Classifier
GFC	Gabor Filter Classifier
LDA	Linear Discriminant Analysis
OGPCI	Oriented Gabor Phase Congruency Image
OOB	Out-of-bag PBGFC Phase Based Gabor-Fisher Classifier
PCA	Principle Component Analysis
RGB	Red Green Blue
RF	Random Forest
ω	Center frequency of sinusoidal plane wave
θ	Orientation or normal direction
σ	Standard deviation along x- and y- direction of Gaussian envelope
ψ	Feature vector
I	Input image.
μ	Mean image
w_i	Mean centered image
C	Correlation matrix of training data
d_i	Face descriptor
\emptyset	Gabor wavelet

LIST OF APPENDICES

APPENDIX A: COMPUTER SPECIFICATION	93
APPENDIX B: MATLAB CODES	94
APPENDIX C: USER INTERFACE	98

CHAPTER 1

INTRODUCTION

1.1 General Introduction

- The commonly used biometrics in criminal investigation applications has many weaknesses (Datta, Datta and Banerjee, 2015). Iris recognition has high accuracy (Hassan, 2017), but it is cost ineffective for implementation on large scale and the surveillance cameras with ultra-high resolution are needed (Datta, Datta and Banerjee, 2015). Fingerprints biometrics are reliable, accurate, non-intrusive and widely and positively received by people (Datta, Datta and Banerjee, 2015), but if individual is non-corporative, this method faces difficulty. Face recognition can work in unrestrained information extraction conditions and acquire large information from unaware people. It balances between reliability and social acceptance compared with iris recognition and fingerprint biometrics.

Face detection and recognition becomes a recently popular biometrics analysis in multitude of criminal investigation applications. It is a nonintrusive and inconspicuous detection method in recognizing people. Face detection and recognition are a remarkable and crucial ability that human beings possess to make a personal identification in daily lives. Thus, the application of face recognition in criminal investigation is an area worth pursuing.

1.2 Importance of the Study

The impetus in developing computer vision based automated facial recognition comes from the criminal investigation aspects in the more connected and networked world today in which the identity verifications are sought after at the places such as the commercial area or the public arena. For example, the assassination of Kim Jung-Nam was taped by the surveillance cameras installed in the airport. The face recognition system matched the suspect's face appeared on the footage to a featured picture of a woman on a webpage of talent show (CNBC, 2017).

The face recognition works differently in different application scenarios. The first one is called recognition or identification. The second one is called authentication or verification. For both scenarios, face images of known individuals are stored in the system earlier and the set becomes a gallery. The incoming images of person at latter time become probes to compare with images stored in the gallery. In criminal investigation, the first scenario is used where the matching is one-to-many and a probe is compared with entire images of the gallery to generate the best match that obtains the highest probability score.

1.3 Problem Statement

In criminal investigation field, a surveillance camera could capture the face of the suspect, either the whole or the partial view of face. Although there are more and more public surveillance cameras installed in the country, the idea of tracking down the person of interest using the on-the-street public surveillance cameras is still less popular.

The examiners in law enforcement agency could skip the lengthy suspect lists using the facial recognition system to narrow it down to a fewer number of people. Even though the surveillance cameras are becoming more prevalent in the public space in recent years, the quality of the image captured still poses the challenge in face recognition as the image could be blurry, the illumination could be inadequate or the suspect's face might not appear as completely frontal on camera screen.

The difficulty of procuring large facial image datasets for the same individuals is also a major concern in criminal investigation. Therefore, a face recognition algorithm that does not require large training datasets is required.

Moreover, human face is not rigid and not the same always. This is because it is subject to many factors that causes variation of its appearance. The two main reasons of variation in human faces can be grouped into two categories: intrinsic factors and extrinsic factors.

Intrinsic factors include the varying facial appearance of same person due to facial expression. The extrinsic factors include the interaction of light with the face such as brightness level, head scale and orientation, resolution and noise. These

factors bring the difficulties in face detection and recognition that degrade the performance.

Another challenge arisen during face recognition is that there exists the dimmed lighting condition or over-bright condition which renders the face underexposed or overexposed (Kong and Zhu, 2007). Performance will be impaired noticeably at large fluctuation in illumination conditions (Datta, Datta and Banerjee, 2015). The facial expression change can also impair the recognition rate.

In controlled environment, the face images are usually frontal views and illuminated by controlled amount of fixed positioned light source. These conditions impose restrictions on the patterns acquired from face images. However, face patterns in real life are beyond the controlled conditions and recognition performance is often impaired compared with controlled environment. The examples of uncontrolled condition include illumination or pose variation and expression changes. The face recognition algorithm has to take into account of those uncontrolled parameters which makes the system more complex.

The aforementioned problems become the core study of development and evaluation of face recognition technique in an effort to overcome the constraints posed by problems above.

1.4 Aims and Objectives

The aim and objective of this project is to implement a method for face detection and recognition by using the fusion between Gabor and Maximum Response filters combined with Random Forest classifier in uncontrolled environment. The mentioned technique is believed to be able to detect, differentiate and recognize facial images with varying lighting conditions or with pose variations which can further assist the law enforcement agency in narrowing down the candidate list of suspects during criminal investigation (Struc and Pavesic, 2010).

Recognition rate and Receiver Operating Characteristics curve will be deployed as the measurement of performance of the proposed technique on unconstrained environment databases which are Labelled Faces in the Wild (Learned-Miller, et al., 2016) and Unconstrained Facial Images databases (Lenc and

Král, 2015). The performance of the implemented technique will be evaluated by comparison with existing face recognition methods.

1.5 Scope and Limitation of the Study

This project only concerns the development of a technique to detect and recognize the face of different people from image databases under uncontrolled environment. The databases are to be obtained from the online open resources and they are to be used during the standard benchmarking. Facial images with differences in head scales and positions and different brightness conditions are part of the database contents.

The factors of speed of recognition and the real time recognition are excluded from the scope of this project due to the limited computation resources.

1.6 Contribution of the Study

There are many publications on the face detection and recognition. This project intends to apply a simpler and less computational demanding algorithm in face recognition system using Gabor-Maximum Response filters with Random Forests classifier. The implemented algorithm differs from the much lauded deep neural networks for which deep neural networks require much larger training datasets than the implemented method. In reality, large-volume of facial image datasets of the same individuals are difficult to obtain. The feature selection using MC-UVE-PLSR is included to trim the large dimensionality of feature sets to save computation costs while maintain the accuracy rate (Han, et al., 2008.). The implemented method is applied on LFW and UFI databases to test the reliability of the face recognition system. The algorithm performance is compared with the existing methods in terms of recognition rate and ROC area under the curve level.

1.7 Outline of the Report

The introduction of the report in Chapter 1 explains the problem issues encountered and the aim and objectives of the project.

The literature review in Chapter 2 highlights the related works about the feature extraction methods, image pre-processing tool and feature classifiers.

The methodology in Chapter 3 explains about the experimental setup and procedures in performing the image processing and machine learning.

The result and discussion in Chapter 4 shows the result obtained and the discussions entailed. Graphs and tables were generated to visualize the data.

The conclusion and recommendation in Chapter 5 concludes the performance of the implemented face recognition algorithm and future recommendations are included for future reference.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

2.1.1 Face Recognition System

In the face processing system, the input face image will undergo a few phases as demonstrated in Figure 2.1.

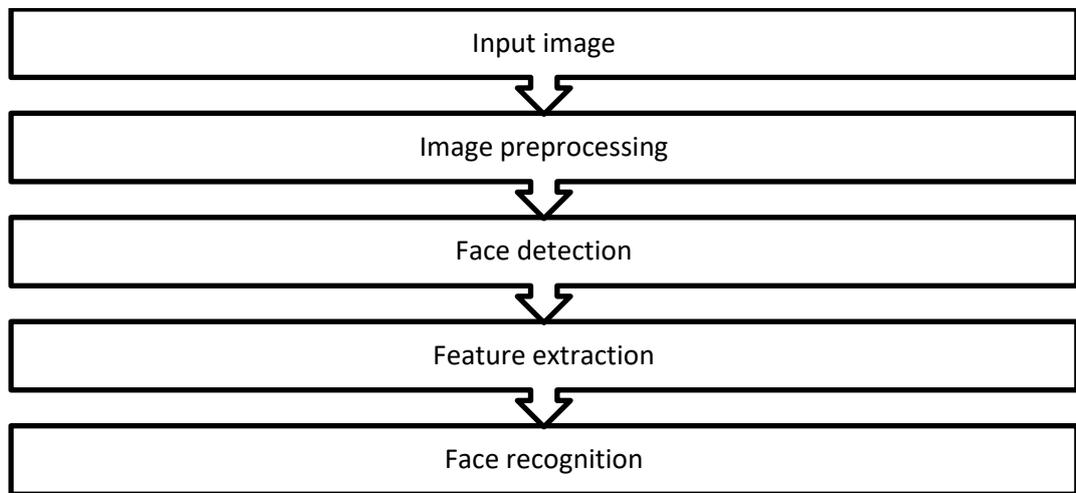


Figure 2.1: The Phases of the Face Recognition System

The input image is either the static image from the image database or the video frame. The preprocessing performed on the input image is to reduce the computation time by converting the RGB image into grey-scaled image or by resizing the image to lower resolutions. The face detection is done by scanning the image to elicit the face region and discard the unnecessary information such as car, furniture and building. The removal of unnecessary information decreases the image dimension and cuts down the computation time. Only face region is retained for next processing. The face region contains the features of eyes, nose and mouth.

The feature extraction stage chooses the characteristics of face region. The high dimensional original face region is reduced to a set of lower dimensional data known as features. The subset of features that is best describing the facial characteristics is chosen for training the classifier and carrying out the classification in the recognition stage.

The face recognition classifies the face according to the designated classes. The most distinguishing features will be used to differentiate the characteristics among every class members. Training or learning method is carried out to make classification of the face regions. The face region is labeled based on the matching score. The accuracy of recognition is measured and evaluated by plotting the ROC curve.

2.1.1.1 Illumination Normalization

2.1.1.1.1 Tan-Triggs Illumination Normalization

The purpose of preprocessing normalization is to minimize the adverse effects of illumination variations, shading or flashing of the images while still retaining the important features in recognition process. There are three steps involved in this normalization recommended by Tan and Triggs. (2007).

A) Gamma Correction

This non-linear image transformation converts greyscale level I into I^γ where γ ranges between 0 and 1. It serves to broaden the dynamic range of image under shades or to narrow down the dynamic range of image under bright condition. This is to restore and retain the image information independent of illumination. The γ value is set to 0.2 (less than 0.5) to avoid over-amplify unnecessary noise in shadowed regions.

B) Difference of Gaussian (DoG) Filter

Gamma correction does not totally eliminate the effect of intensity gradients for example the shades falling onto the subject in the image. The shading due to face surface structure is hard to be differentiated from the shading due to illumination gradients. The shading due to face surface structure is an useful

information about the face features but illumination gradients are not. Thus DoG filter is used to suppress the highest spatial frequencies to minimize aliasing and noise.

DoG filter is a bandpass filter. It has smaller Gaussian width, $\sigma_0 = 1$ pixel and outer Gaussian width with σ_1 of 2 pixels. The outer Gaussian width is set to 2 pixels so that the informative low frequency information is not destroyed indiscriminately. The inner Gaussian width is set to 1 pixel due to the fact that this value to width helps to smoothen the image to reduce noise.

DOG filter is implemented through the convolution between image and filter. If the gamma correction is absent before the DOG filter convolution, the local contrast within the shadowed area in the image will be reduced hence useful face feature information is destroyed.

C) Contrast Equalization

Image intensities are adjusted accordingly such that it standardizes the overall intensity changes. For example, an image has a small portion of signals which has extreme values found in nostril's dark regions.

The global image intensity is then rescaled to standardize the overall image intensities. The reason of introducing contrast equalization is that the image still has a small portion of signals that is of extreme values such as nostril's dark regions. The equalization process is shown as follows:

$$I(x, y) \leftarrow \frac{I(x, y)}{(\text{mean}(|I(x', y')|^\alpha))^{1/\alpha}} \quad (2.1)$$

$I(x, y)$ is the pixel intensity of coordinates (x, y) ; α is the parameter that reduces the extremity of the image intensity. By default α is set to 0.1.

The resultant image might still have extreme intensity. To suppress this extremity, the non-linear function is applied. The non-linear function is shown as follows:

$$I(x, y) \leftarrow \tau \tanh\left(\frac{I(x, y)}{\tau}\right) \quad (2.2)$$

τ is the threshold to get rid of high values after the normalization. By default $\tau = 10$. The resulting image intensity is restricted to the range $(-\tau, \tau)$.

2.1.1.2 Feature Extraction

The purpose of feature extraction is to elicit relevant information from the input image and to downsize the data of high dimensionality training image. The result of feature extraction is a set of information known as features projected in subspace with lower dimensionality. Unique features are chosen while the excessive or irrelevant features are rejected. Extracted features are commonly projected into subspace. The features that are most distinguishing are used to in the next process which is training of classifiers.

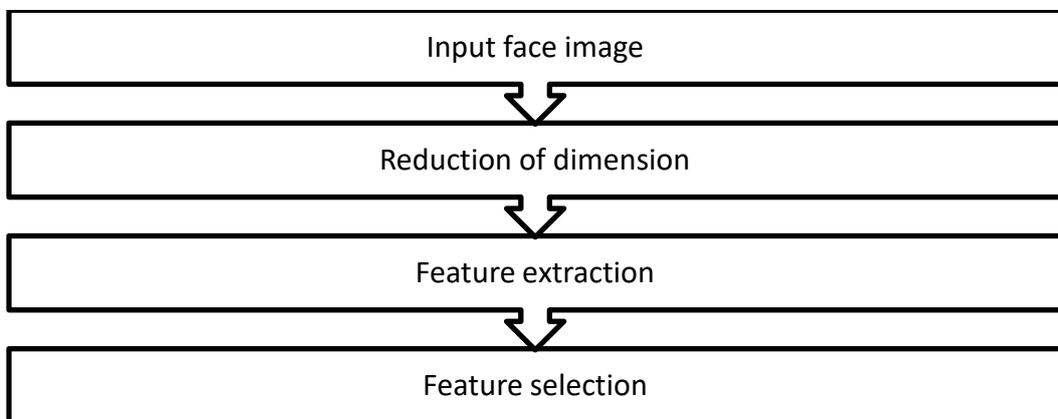


Figure 2.2: Feature Extraction Stage

The feature extraction process has three parts: downsizing of dimensionality, feature extraction and feature selection. The downsizing of the dimension reduces the high dimensional input data to a lower scale known as features. The feature extraction is implemented by convolving the input data with a filter and the output of the filtering would be the features. The most useful feature subset will be used for further processing. Figure 2.2 demonstrates the work flow of the feature extraction process.

2.1.2 Face Detection Method

Viola and Jones face detection method is able to carry out the classification of face images according to the value of simple features. This feature-based system can work faster than the pixel-based system. The simple features are based on Haar basis functions. The Haar-like features is a scalar product between the input face image and the Haar-wavelets (Viola and Jones, 2004).

According to Sun et al. (2016), there are three types of Haar-like features. Firstly, a two-rectangle feature is the difference between the total amount of pixels of two rectangular regions as shown in Figure 2.3 (a) and (b). Secondly, a three-rectangle feature is the difference between two side rectangles and the middle rectangle sandwiched in between the two side rectangles as shown in Figure 2.3 (c) and (d). Thirdly, a four-rectangle feature is the difference between two pairs of rectangles which are placed diagonally as shown in Figure 2.3 (e).

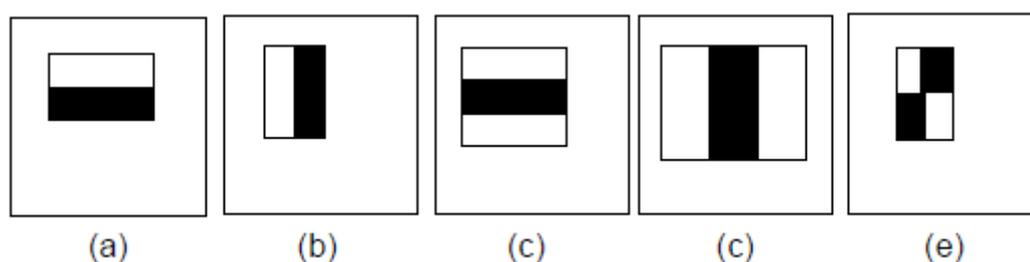


Figure 2.3: Five Haar-like Patterns (Datta, Datta and Banerjee, 2015)

By using integral image, rectangular features of Haar-like features can be generated at a higher speed. The integral image at row and column positions, (i, j) has the total amount of pixels situated on top and on the left hand side of (i, j) . Under integral image, a single feature can be analyzed at any scale and any position in a few operations which allows the face detection process to be done over the whole image at faster pace.

In Figure 2.4, the value of integral image at position 1 is the total addition of the pixels in rectangle A. Value at position 2 is the total addition of the pixels in rectangles A and B. For position 3 and 4 they are $(A+C)$ and $(A+B+C+D)$. In Figure 2.5, the sum of pixel values within shaded region (Heyden, 2006) is given by:

$$Sum = I(D) + I(A) - I(B) - I(C) \quad (2.3)$$

where

I is the integral image.

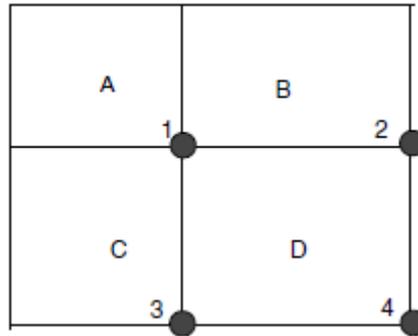


Figure 2.4: Integral image



Figure 2.5: Two-rectangle feature from integral image

Next, the Viola-Jones face detector uses AdaBoost to choose a subset of features to train the classifier. AdaBoost is composed of weighted sum of multiple weak classifiers (Zhu et al, 2009). Each weak classifier acts as the threshold on Haar rectangular feature. Weight is assigned based on the correctness of classification made by weak classifiers. Then these weak classifiers are combined to form a complex cascade to spend computation on the image regions having the possible presence of faces. Cascade classifier calculates the probability of presence of face in each sub-window image. Those sub-windows which are confirmed the absence of face image are immediately discarded. The sub-window which passes through the entire stages of cascade classifiers is confirmed containing a human face.

2.1.3 Feature Extraction

2.1.3.1 Local Binary Patterns (LBP)

Local Binary Pattern is one of the method to discriminate the textures and edges within an image (Ojala, et al., 2002). The LBP kernel operates on the change in intensity in the neighbourhood of a pixel. The kernel designates value of 0 or 1 to the pixels in the neighbourhood:

$$value = \begin{cases} 0 & \text{when neighboring pixel} < \text{centre pixel} \\ 1 & \text{when neighboring pixel} \geq \text{centre pixel} \end{cases}$$

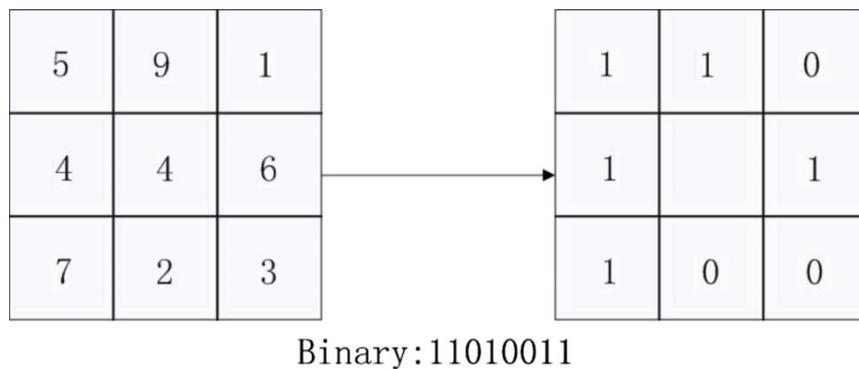


Figure 2.6: Visualization of LBP

The transformation of original pixel values into LBP pixel values is shown in Figure 2.6. The LBP kernel can also be modified into using circular neighbourhoods with different radius and number of points. The points are evenly distributed along the circumference of the circle and the same threshold comparison process is carried out as shown in Figure 2.7. The points which are not located at the center of a pixel is bilinearly interpolated to obtain the pixel value.

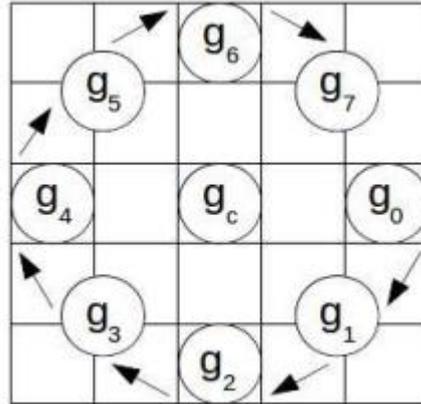


Figure 2.7: Circular LBP

The histogram of LBP values are used as the facial feature representation. However, the LBP based feature extraction carries out histogram computation on uniform and predetermined grid in the facial image and does not consider the properties of a particular image. For example, there is a situation where there is large difference of positions of facial features between images with large pose variations but LBP still considers the fixed coordinates and number of facial fiducial points (Lenc and Kral, 2016). This could not give most representative feature points of a face.

2.1.3.2 Gabor Filters

Gabor filter is used in feature extraction. Gabor filter is able to extract the features in spatial and frequency domains and it is robust to illumination variation and expression changes (Bargavi and Santhi, 2014). According to Daugmann (1980), the 2D Gabor filters can perform approximately like simple cells in human visual cortex. These characteristics can be used to elicit the features aligned at specified orientation. Gabor wavelet coefficients consist of different scales and different orientations. Gabor filter can elicit important visual features like spatial localization, orientation selectivity and frequency selectivity to generate a filtered output image with peak intensity points. The filter function is a combination of sinusoidal plane wave and Gaussian function.

The Gabor filter kernel, φ is represented by equations as follows (Abdulrahman, 2014):

$$\varphi(x, y, \omega, \theta) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} e^{i\omega x} \quad (2.4)$$

$$x' = x \cos \theta + y \sin \theta; \quad (2.5)$$

$$y' = -y \sin \theta + y \cos \theta \quad (2.6)$$

where

(x, y) is the pixel location

ω is central frequency of sinusoidal wave

θ is the orientation or normal direction to the sinusoidal plane wave

σ is the standard deviation along x - and y - direction of Gaussian envelope which also defines the radius of Gaussian.

According to Vitomir and Nikola (2010), it is recommended that the Gabor filter bank to be set at forty banks at five scales and eight orientations with standard deviation at 2π and center frequency at $\sqrt{2}$ Hz. The Gabor filter output is shown in Figure 2.8.

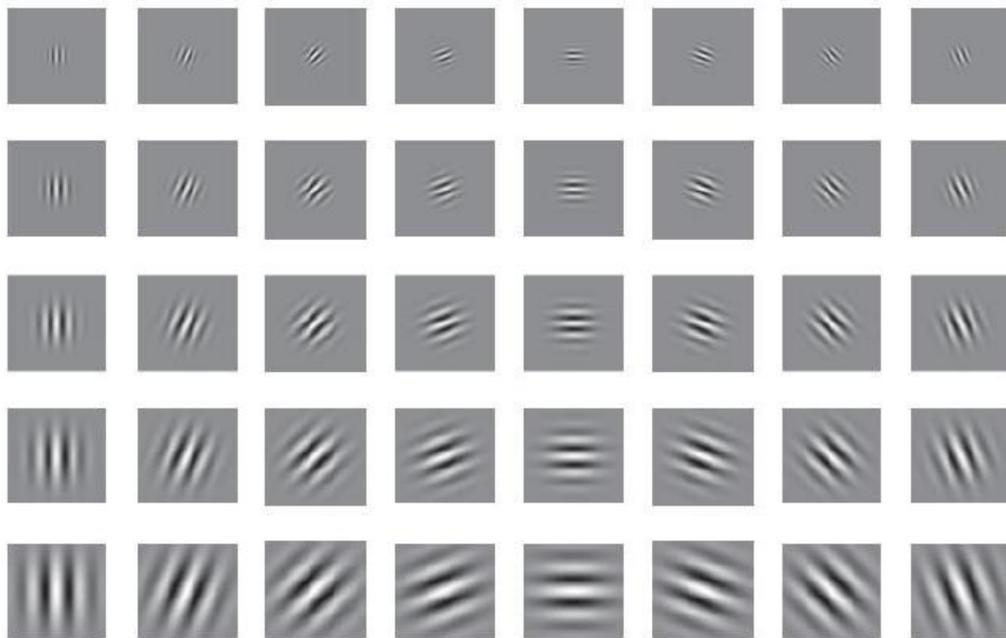


Figure 2.8: Forty Gabor filters with five scales and eight orientations

The feature representation is obtained by convoluting the Gabor filter bank with the image:

$$\psi(x, y) = I(x, y) * \varphi(x, y, \omega, \theta) \quad (2.7)$$

where

$\psi(x, y)$ is the feature vector

$I(x, y)$ is the input image

$\varphi(x, y, \omega, \theta)$ is the Gabor kernel

Gabor filters is the Gaussian function modulated by sinusoidal waves at the center frequency, f_u and orientation, θ_u . The feature representation has the wavelet coefficients that varies the scales and the orientations which make filter representations more invariant to translation, rotation, distortion and scaling. The extracted feature vectors are combined into one single feature vector.

The output of convolution is a complex value, having a real part, $H_{m,n}(x, y)$ and an imaginary part, $K_{m,n}(x, y)$.

$$H_{m,n}(x, y) = \text{Real}[\varphi_{m,n}(x, y)] \quad (2.8)$$

$$K_{m,n}(x, y) = \text{Imaginary}[\varphi_{m,n}(x, y)] \quad (2.9)$$

$$A_{m,n}(x, y) = \sqrt{H_{m,n}^2(x, y) + K_{m,n}^2(x, y)} \quad (2.10)$$

$$\theta_{m,n}(x, y) = \arctan\left[\frac{K_{m,n}(x, y)}{H_{m,n}(x, y)}\right] \quad (2.11)$$

Where

$A_{m,n}(x, y)$ is the amplitude

$\theta_{m,n}(x, y)$ is the phase information of the output of Gabor filter.

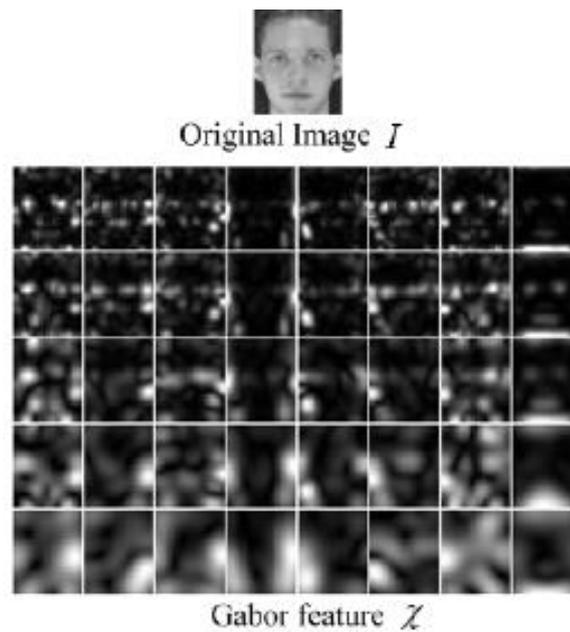


Figure 2.9: Gabor Representation of Facial Image (Zhou, Yin, and Zhang, 2011)

Figure 2.9 shows the Gabor representation of facial image. The magnitude of the output is used for face recognition as it does not change dramatically with positional displacement (Liu, 2001). Phase is too sensitive even at minor displacement (Zhang et al, 2007). The phase information of the feature is unstable. This is the reason the phase information is rarely used in feature extraction. It is onerous to elicit stable and pronounced features from phase information (Zhang et al, 2007). Figure 2.9 shows the Gabor representation of facial image.

The oversensitivity and instability of phase information can be tackled by the formation of histogram of Gabor phase pattern (HGPP) (Zhang, et al., 2007). The HGPP is established through local binary method.

According to Struc et al. (2008), the Oriented Gabor Phase Congruency Image (OGPCI) was proposed to extract phase feature information without much severe compromise to the stability. Struc and Nikola (2010) integrated Gabor magnitude and phase feature information with Linear Discriminant Analysis (LDA) to devise a new method known as complete Gabor-Fisher classifier. Both authors claimed that the method performed better than Principal Component Analysis.

During feature extraction, the Gabor magnitude information is obtained first. Each response has the same dimensionality as input, thus the pixel space is expanded to 40 times of the original size. The reason of the expansion is that the Gabor filters are not orthogonal to one another and this leads to information redundancy. The magnitude information can be downsampled and dimensionality can be reduced using feature selection method such as principal component analysis.

Compared with LBP, Gabor filter has an advantage that it can automatically detect the positions of facial feature points. By using Gabor filter, local extrema are located and will be used as feature points. These feature points are not singlehandedly confined to certain positions within the predefined grid as seen in LBP. Thus, Gabor filters are better than LBP in discriminating between similar shaped faces (Xia, 2013).

2.1.3.3 Maximum Response Filter

Maximum Response Filter (MR) is derived from root filters which are made up of 38 filters. Both of the Gaussian and Laplacian of Gaussian (LOG) filters are used. Compared with Gabor, MR filter has additional element which is LOG. Laplacian function helps to detect edges by looking for the zero crossings in image. The combination of Laplacian and Gaussian functions is due to the reason that Gaussian kernel helps smoothening the image to reduce noise before Laplacian function for edge detection (Caenen, 2004).

The edge and bar filters in MR filter have three scales for each. Edge and bar filters have six orientations for each scale level. MR filter has also two isotropic filters. Each of the two isotropic filters gives a single response. For edge, bar and isotropic filters, the maximum filter response of all orientations is obtained.

MR filters are used as it could respond to oriented image patches and anisotropic textures. The presence of isotropic and anisotropic filters in several orientations generates useful texture features. The MR filters can also detect the angle of maximum response. This functionality helps in discriminating among those similar textures. The visual representation of MR filters is shown in Figure 2.10.

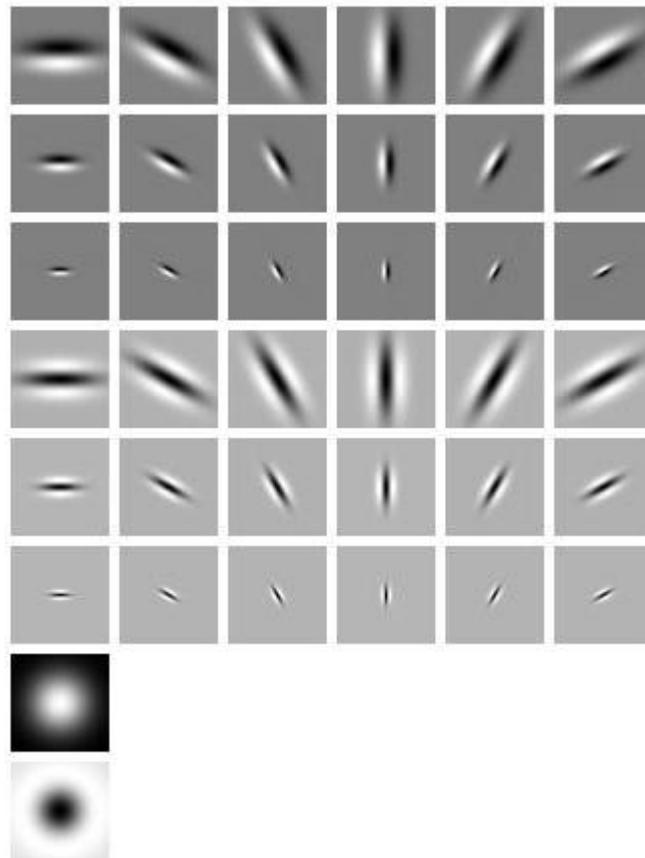


Figure 2.10: Maximum Filter Response (Caenen, 2004)

2.1.4 Feature Selection

Feature selection is to select the best subset of features that brings smallest error in classification with lesser data and lesser complexity (Brownlee, 2017). Smaller classification error leads to higher classification accuracy. The feature selection is carried out analysing every subset of features and selecting the subsets that meet the pre-determined criterion. Feature selection differs from the dimensionality reduction in which dimensionality reduction produces a totally new combination of variables while the feature selection only includes or excludes certain existing variables in the dataset without making structural modification to the dataset (Wang, 2015).

There are three feature selection methods:

(a) Filter methods

This method uses statistical measure to designate a score to every feature and rank them according to the score (Brownlee, 2017). It uses indices (such as regression coefficients or variable importance in projection) along with thresholds to filter away irrelevant predictors. It measures the feature relevance using statistical tests. However, its drawback is that it could fail to detect the features with highest reliability.

(b) Wrapper methods

This method measures the usefulness of feature subset (Kohavi and John, 1997). Wrapper method is similar to filter method but differs in the aspect of threshold determination. In wrapper method, the threshold of indices is determined using cross-validation. Wrapper method utilizes the search algorithm to obtain the variable subsets, depending on different threshold levels. A model is constructed for each subset and is evaluated by fitting a mathematical model to them. The ultimate model is chosen based on the best prediction performance (Mao, Cai and Shao, 2013).

The example of wrapper methods would be genetic algorithm with partial least square (GA-PLS), Monte-Carlo uninformative variable elimination with partial least square (MC-UVE-PLS) and Competitive adaptive reweighted sampling with PLS (CARS-PLS).

Compared with GA-PLS, MC-UVE-PLS is less complex and less sensitive in tuning the parameters which makes it less complex. GA-PLS requires a lot of user-defined tuning parameters and it is computation exhaustive due to the tediousness in assessing the fitness function which highly depends on the sample size (Mehmood, 2012). Meanwhile, CARS-PLS is highly sensitive in training data selection. Wrapper methods are claimed to give better performance than filter methods (Reunanen, 2003).

(c) Embedded methods

This method is similar to wrapper method but differs in that the feature selection is carried out concurrently with the training phase. The learning algorithm is updated on par with the feature selection. Embedded methods utilize the internal parameters of classification model to carry out variable selection. The drawback of this method is that these concurrent updates require more complicated computation compared to the wrapper methods (Brownlee, 2017).

2.2 Face Recognition

Face recognition is the stage where face regions are classified into correct label of class. There are several categories implemented in face recognition process: appearance-based, feature-based and learning-based methods.

2.2.1 Subspace-based Method

In subspace-based method, the input face image is compressed to fewer dimensions and maximum variance between each orthogonal subspace direction is preserved. Face images can be described using lower dimension subspaces.

2.2.1.1 Principal Component Analysis (PCA)

PCA is firstly used by Turk and Pentland (1991). A subset of principal directions is generated from the face images that are to be used for training. The PCA is to minimize the correlation between different principle axis and to reduce the number of redundant filters.

Projection of subset is made onto the space of the principle components to extract the feature vectors. The two dimensional image matrix is converted into one dimensional image vector. The eigenvectors and eigenvalues of covariance matrix will be computed.

The mean image, μ of N number of training image samples x_i is denoted by:

$$\mu = \sum_{i=1}^N x_i \quad (2.12)$$

The mean centered image, w_i is denoted by:

$$w_i = x_i - \mu \quad (2.13)$$

Let W be a matrix consists of column vectors w_i arranged side by side. The correlation matrix of training data, C is given as follows:

$$C = W^T W \quad (2.14)$$

The eigenvector, E_i of correlation matrix, C is obtained. Eigenvectors are orthogonal and lie in the span of vector of the Gabor filters. This means the eigenvectors are linear combinations of input data. The eigenvectors are similar to the original Gabor filters. The eigenvectors that correspond to greatest eigenvalue show the highest variance in the image. The variance associated with highest eigenvalue exhibits the exponential decrease toward the variance associated with smallest eigenvalue. This means that 90% of the variance is contained within 5% to 10% of the dimensions.

The eigenvalues of the correlation matrix, v_i is denoted by:

$$v_i = \frac{1}{N} \sum_{j=1}^N (E_i^T w_j)^2 \quad (2.15)$$

The eigenvalues enables the eigenvectors to describe the variation between the training set images. Each eigenvector contains position information of an image.

PCA enlarges the variation in subspace, and clusters the information with similar weight values. The face descriptor, d_i is denoted by:

$$d_i = E_i^T w_i \quad (2.16)$$

The face descriptor is also known as basis set of images or eigenface. It has the identical dimension with the training face images. The basis set images known as the eigen images have the greatest eigenvalues and contain most of the information of the training set images. Each image in the training set is approximately the linear combination of eigen images.

Turk and Pentland (1991) proposed to use the Euclidean distance between the feature vectors and make the comparison between the distances to carry out the face recognition. The similarity between input and training images is evaluated through the distance calculation. If the distance is greater, it means that the matching between the two sets of images is lower. However, PCA performs better for frontal face image but poorer if the face is angled according to Turk (1991).

2.2.1.2 Linear Discriminant Analysis (LDA)

In PCA, the direction of greatest variance might not helpful in classification process as it may not have discriminative information. It is the illumination variation that causes much of the variation in data. PCA does not differentiate the differences between classes as extensively as LDA.

Belhumeur et al. (1997) proposed to use LDA method in face recognition. Under LDA, the face image features are represented using scatter matrix. LDA makes the data of different classes further from each other and clusters the data of the same class closer to each other. This maximizes the ratio of between-classes scattering to within-class scattering. The eigenvectors are obtained from the scatter matrix and Euclidean distance measurement is used to compute the similarity between input image and the training image.

Unlike PCA, LDA needs bigger samples for training at higher accuracy. According to Zhao et al. (1998), the merge of PCA and LDA has the capability to reduce the size of training and preserve the advantage of class discrimination.

2.2.1.3 Geometric Feature-based Method

Geometric feature-based method elicit the face features for instance mouth, nose and eyes to identify their position in the image, the geometrical shape and distance and orientation among these features. The face feature parameters such as width, length, shape and location are used by classifier to carry out the recognition process by measuring the Euclidean distance between the feature vectors. The geometric feature based method has an advantage of accommodating the flexible deformation at feature points which makes the classification robust to face image with pose variation. Figure 2.11 shows the measurement of the face features using geometric feature-based method.

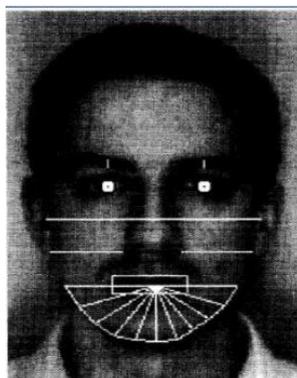


Figure 2.11: : Geometric Feature-based Method

Sharif et al (2011) suggested the application of Gabor wavelet in identifying the position of the face features. After Gabor filter is applied to the input face image, the peak intensity points (feature points) of the filter output image is attained. The feature vectors of the input image are used to make comparison with the feature vectors of training image by measuring the mean distance of the feature vectors between the input image and the training image. The smaller the mean distance, the higher the matching score.

2.2.2 Learning-based Method

2.2.2.1 Neural Network with Gabor Filters

Artificial neural network was first proposed by Kohonen (1988) to carry out the recognition of aligned and normalized faces. The neural network based methods are used for classification and also for feature extraction.

Bhuiyan et al (1997) suggested neural network method combined with Gabor filter [10]. Each image undergoes normalization of contrast and illumination. The noise is filtered by median filter and mean filter. The image is fed into Gabor filter. The 15 Gabor output images (five orientations and three scales) contain the variation information measured by the filter. The neural network first layer gets the Gabor features. The number of nodes is designed to be equal to the dimension of Gabor feature vectors. The output is the number of images that shall be recognized by the program. The training employs back propagation algorithm as follows:

1. Initialize the weights and threshold
2. Reiterate the process until terminal criterion is achieved.
 - a. Apply input and obtain the desired outputs. The actual outputs of neurons is calculated using sigmoid activation function which gives output from -1 to 1 with threshold at 0 for detecting a face.
 - b. The weights are updated. The errors are propagated backward.
 - c. The iteration count is increased

However, neural network training needs exhaustive training and reiterates many times on the training images. This takes long period and huge memory storage. The network may train on images batch by batch and lose track of the characteristics of first image. The equal number of negative examples and positive examples are used in training which may not be feasible in real life situation.

2.2.2.2 Support Vector Machine (SVM)

Support vector machine (SVM) is introduced in Statistical Learning Theory (Cortes and Vapnik, 1995). Assume that a dataset D consists of m pairs (x_j, y_j) where x_j is real number and y_j is a binary value of either -1 or +1. The SVM conducts its search for a hyperplane to divide the data in a maximized margin of linear separation between the classes with minimum error. The hyperplane is the classification boundary that separates the blue and red feature points with maximum distance margin in Figure 2.12. The mapping function in SVM maps the dataset D to a space with greater dimensionality to allow for the linear separation between classes. The mapping function could be in the form of Gaussian or Radial Basis functions (Lorena, 2011). The drawback of SVM is that it requires extensive user-defined parameter adjustment.

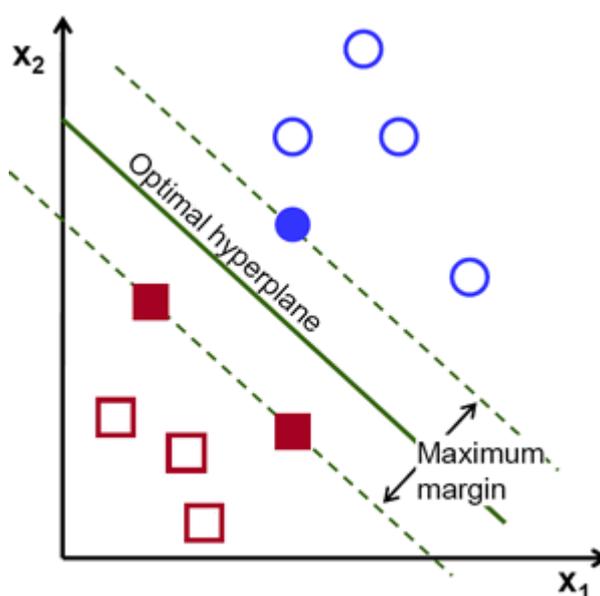


Figure 2.12: Support Vector Machine (Docs.opencv.org, 2017)

2.2.2.3 K-Nearest Neighbours

In k-Nearest Neighbour (kNN), the new data point is classified with regard to the class of the majority of its k nearest neighbouring training data points (Lorena, 2011). In other words, the Euclidean distance between the data is calculated and compared against. The strength of kNN is that it is simpler compared with SVM. However,

the prediction duration it consumes is expensive as it has to revisit the entire training data (Lorena, 2011).

In Figure 2.13, the distance between new data point, x_j and the 5 nearest neighbouring data points are measured. x_j is belonged to a cluster ω_1 as the majority of 5 nearest neighbouring data points falls within the cluster ω_1 . The strength of kNN is that it is simpler compared with SVM. However, the prediction duration is longer as it has to revisit the entire training data.

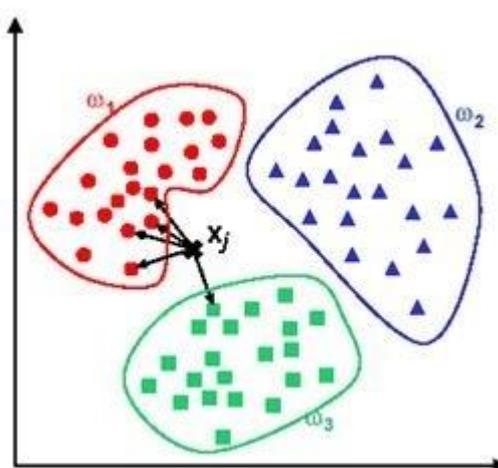


Figure 2.13: K-Nearest Neighbours (Srivastava, 2017)

2.2.2.4 Random Forests (RF)

Random forest uses bagging method and random feature selection to build a decision trees to reduce the variance and to avoid overfitting (Breiman, 2001; Ho ,1998). Bagging is also known as the bootstrap aggregating. Base learners are built sequentially in boosting; while base learners are built in parallel in bagging.

A forest is constructed from many independent classification trees. Randomness is created from the dataset by selecting the data subset randomly. The trees are made different by having the training carried out on slightly different data subset, and bootstrap samples can be taken from dataset for each tree. Bootstrap sample is the sample taken from original dataset with replacement. It is the same size as the original dataset. This creates a set of classification learners that function

slightly differently. Some data is present more than once in the sample set and some data does not appear at all in the sample.

According to Breiman (2001), the probability of i -th training data being selected at least once is $1 - \frac{1}{e} = 0.632$. This implies that there are 36.8% of training data that is not used in training process. Under random forest, the node is split based on the best among those randomly chosen subset of features at that particular node.

The model is fitted to each dataset, and combines them by assigning the output as the majority vote of all classifiers. Besides that, the choices that are to be made by decision trees are limited. A random subset of features is fed to the tree, they can be only selected from that particular subset instead of from entire set. At the end of the tree, the similarity is divided by number of trees and a matrix is formed where each element has value within unit interval $[0, 1]$. During every observation, every individual tree will be voting for one class. The random forest will make the prediction based on the result of the class that commands the highest votes.

The training phase of Random Forest has higher efficiency than Bagging because Bagging evaluates the entire features for split decision while Random Forest only evaluates a feature subset. Random forest is user friendly (Kouzani et al., 2007) that only two parameters need to be tuned. These are the number of features in the random subset at the node and the number of trees.

When randomness in training of every tree is increased, the training process is sped up as there will be lesser features to look for at each stage. The size of subset is the square root of number of the features. The purpose of randomness is to reduce the variance without affecting the bias. The tree trimming is not needed.

The trees do not rely on each other, therefore decision can be obtained from different trees on different individual processors. Random forest can be executed on as many processors with linear speed enhancement.

Kouzani et al (2007) employs random forest for image classification using pixel value of image. This method is sensitive to illumination variation nonetheless. Ghosal et al (2009) merges Gabor filter with random forest to create stronger classification to solve the problems faced by Kouzani's method.

According to Lorena (2011), RF outperforms the SVM and kNN. Besides that, it is also claimed that RF gives stability of performance with lower standard deviation of RF ranking.

2.3 Summary

In short, the Gabor and Maximum Response (MR) filters which give global representation of face features, is more suitable in for face feature extraction under uncontrolled environment than Local Binary Pattern (LBP) as LBP focuses on small patch of pixel. The Random Forests (RF) classifier avoids overfitting due to randomness of subsampling during the construction of decision trees. Random Forests has fewer parameters for tedious adjustment than Support Vector Machine. Therefore, Gabor and MR filters are used in feature extraction and Random Forests is used as classifier. The details will be discussed in Chapter 3.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

The proposed system is divided into two main phases namely training and testing phases. The flow diagram of these two phases are shown in Figure 3.1 and Figure 3.2 respectively.

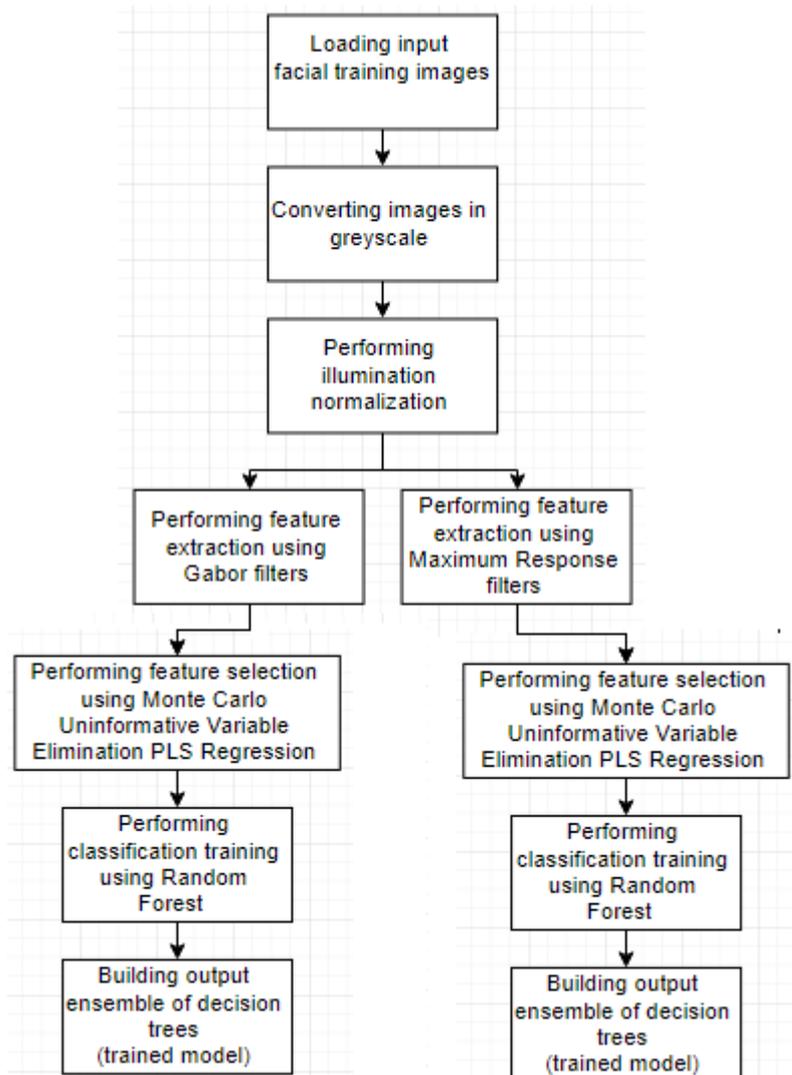


Figure 3.1: Flow diagram of training process

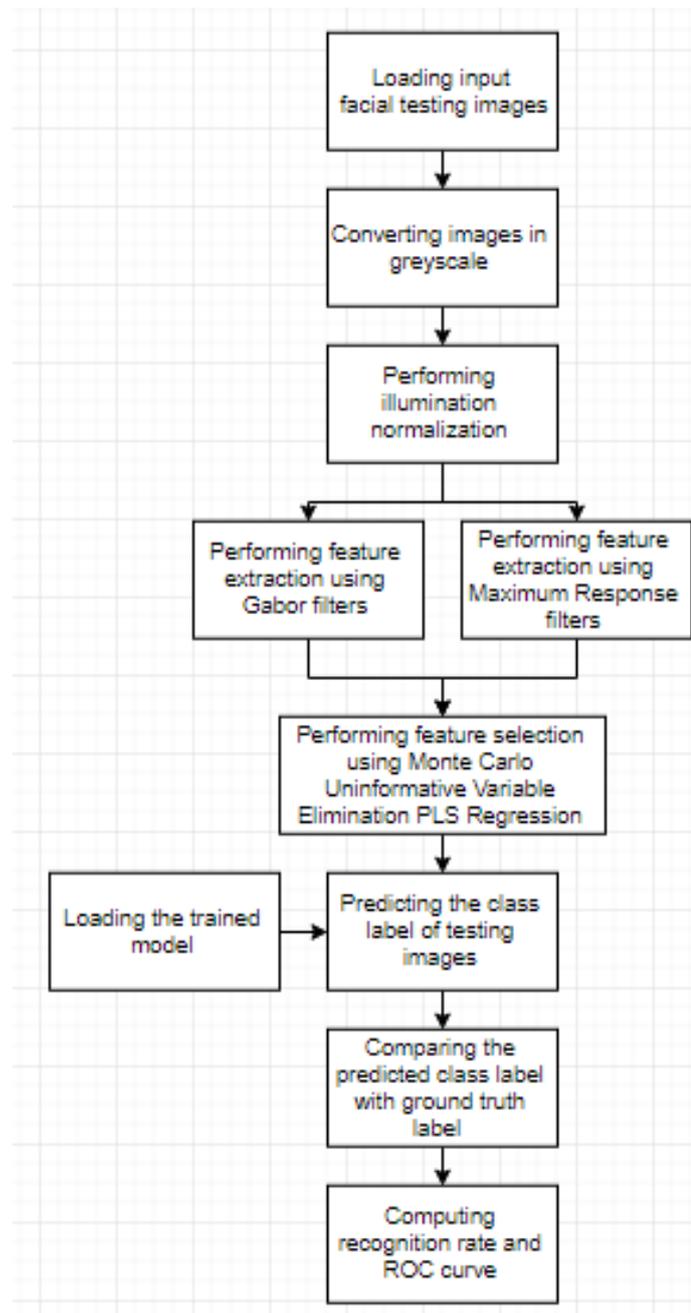


Figure 3.2: Flow diagram of testing process

There are two main branches involved in the program. They are training stage and testing stage. The training stage trains the program to implement classification process of the sampled face image based on the labelled data. The testing stage involves the application of the trained module to carry out the classification of the test image.

In training process, the feature vectors generated are fed into the classifiers using Random Forest algorithms to produce a trained and learned model. The feature vectors are utilized to grow a Random Forest.

In the testing process, the prediction is based on the obtained feature vectors of the testing images using the trained decision trees. The prediction function generates a matrix of matching scores which state the probability of that particular testing image belonging to a specific class. The class candidate with highest matching scores suggests that the testing image is most probably belonged to that particular class.

The performance of random forest constructed is evaluated by analyzing the receiver operating characteristic (ROC) curve. ROC curve is the evaluation graph that investigates the interaction between true-positive-rate and false-positive-rate. The recognition rate is carried to measure the classification accuracy rate. Further information about the evaluation tools will be discussed in Chapter 4.

3.2 Databases

3.2.1 Labelled Faces in the Wild (LFW)

Table 3.1: Experimental Setup of LFW Database

Database	Labelled Faces in the Wild (LFW)
Number of individuals used	57
Images per individual used	20-30
Number of training images	1460
Number of testing images	593

Resolution (pixels)	Between 100×100 and 150×150
Format	24 bit color JPEG
Background	Random objects/scenery, not necessarily with plain color
Pose	Varies
Illumination level	Varies
Expression	Varies slightly

LFW data samples have random or rather arbitrary values of variations of the parameters (such as lighting conditions, expression variations and pose variations). LFW data samples differ from controlled environment databases as LFW has no fixed variations of parameters of pose, lighting or expressions. In controlled environment databases, every facial image is varied in one specific parameter while the other parameters are fixed. However, in LFW, there are concurrent variations in pose, lighting condition and expression within the same image as shown in Figure 3.3. The experimental setup of LFW is shown in Table 3.1.



Figure 3.3: Examples of face images in LFW database

3.2.1.1 Training and Testing Protocol of LFW

The data set is organized into two subsets, one for training and one for testing. The individuals appear in the training and testing subsets are mutually exclusive. This is to avoid data overfitting and overusing.

The final parameters of the feature extraction function and the classifier should be set using only training data subset. The adjustment of parameters on the testing datasets is forbidden to avoid unfair result benchmarking.

During cross validation process, each of the training and testing of the data subsets should be carried out independently.

3.2.2 Unconstrained Facial Images (UFI)

Table 3.2: Experimental Setup of UFI Database

Database	Unconstrained Facial Images (UFI)
Number of individuals used	150
Image per individual used	7-10
Number of training images	1100
Number of testing images	150
Resolution (pixels)	128×128
Format	Greyscaled JPEG
Background	Random objects/scenery, not necessarily with plain color
Pose	Varies
Illumination level	Varies
Expression	Varies slightly

Unconstrained Facial Images (UFI) database gathers the photographs from Czech News Agency (CTK). In this database, the photograph collection is divided into two groups: training sets and testing sets. The illumination condition, individuals' pose variation and face expressions within the image are unconstrained, similar to LFW. The details of the experimental set up of the database is shown in Table 3.2. The examples of face images of UFI are shown in Figure 3.4.



Figure 3.4: Examples of face images in UFI database

3.3 Image Pre-Processing

The input face image undergoes pre-processing in the first place by converting the Red, Green and Blue (RGB) scale into grey scale. The image pre-processing is to refine the quality of an image. Filtering is applied to images to reduce image details to speed up computational process. This is because RGB image contains three channels (Red, Green and Blue). Each channel contains 8 bits which makes up 24 bits in total. Greyscale image has only one channel and represents the intensity levels using 8 bits. Instead of using three color dimensions in a RGB image, the greyscale conversion reduces the image to one dimension for quicker computation process.

The Tan-Triggs illumination normalization is then applied on greyscaled image. Upon completing the conversion process, the size of the image is scaled down to 100×100 pixels in order to quicken the computational process. The image pre-processing flow diagram is shown in Figure 3.5.

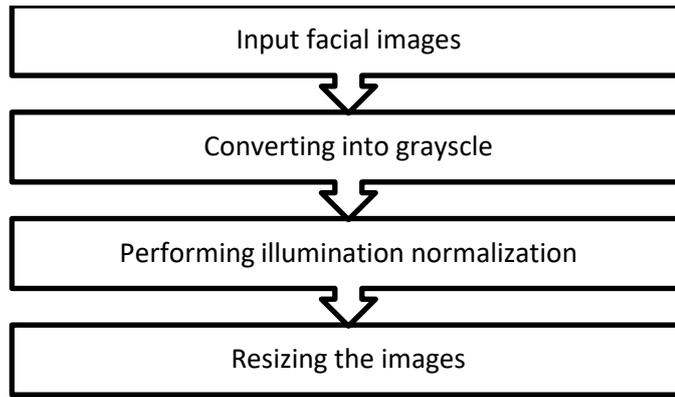


Figure 3.5: Image Pre-processing

3.4 Feature Extraction

The image undergoes feature extraction process using Gabor filters and Maximum Response filters. There are two components of Gabor filters which are Gabor magnitude components and Gabor phase components. The feature vectors generated by the filters are large in dimensionality and consume high computation costs and time complexity. Thus, feature selection is being carried to use top importance ranking features to reduce the number of features for the next stage.

3.4.1 Gabor Filters

Facial features extracted using the Gabor filter has the robustness against illumination and minor facial expression variation. The Gabor wavelet, $\phi_{m,n}(x, y)$.

$$\phi(x, y) = \frac{f_m^2}{\pi A s} e^{-\left(\frac{f_m^2}{A^2}\right)x'^2 + (f_m^2/n^2)y'^2} e^{j2\pi f_m x'} \quad (3.1)$$

Where,

$$x' = x \cos\theta_n + y \sin\theta_n$$

$$y' = -x \sin\theta_n + y \cos\theta_n$$

x and y are coordinates of pixel

f_m = Gaussian center frequency = $f_{max}/2^{m/2}$

m = Number of scales

n = Number of orientations

θ_n = Gaussian orientation = $n\pi/8$

A = Ratio between center frequency

s = Size of Gaussian envelope

f_{max} = Maximum frequency of filter

Struc et al. (2009) recommends that $A = s = \sqrt{2}$ and $f_{max} = 0.2$. Filter bank of five scales and eight orientations is constructed with $m \in \{0, 1, 2, 3, 4\}$ and $n \in \{0, 1, 2, 3, 4, 5, 6, 7\}$. The filter bank has the real and imaginary terms of Gabor wavelet. The real term is used in facial feature extraction process.

The input image is a greyscale image of a face having the size of $p \times q$ pixels. Gabor filter is denoted as $\Phi_{m,n}(x, y)$ with center frequency f_m and orientation θ_n . The transfer function of the filter is the convolution between the greyscale image, $B(x, y)$ and Gabor filter, $\Phi_{m,n}(x, y)$.

$$H_{m,n}(x, y) = B(x, y) \times \Phi_{m,n}(x, y) \quad (3.2)$$

$H_{m,n}(x, y)$ refers to the complex valued output of the filter function. $H_{m,n}(x, y)$ is broken down into real term $R_{m,n}(x, y)$ and imaginary term, $I_{m,n}(x, y)$.

$$R_{m,n}(x, y) = Re[H_{m,n}(x, y)] \quad (3.3)$$

$$I_{m,n}(x, y) = Im[H_{m,n}(x, y)] \quad (3.4)$$

The magnitude of the output filter, $J_{m,n}(x, y)$ is the obtained using Pythagorean theorem.

$$J_{m,n}(x, y) = \sqrt{R_{m,n}^2(x, y) + I_{m,n}^2(x, y)} \quad (3.5)$$

3.4.1.1 Downsampling Method

The features extracted will be large in number when Gabor filter bank of 40 sets is used to process an image. The dimensional size of the image of 100×100 pixels becomes 400,000 ($=100 \times 100 \times 40$), which it is computation exhausting. In order to resolve this challenge, downsampling using rectangular grid technique is recommended by Vitomir and Nikola (2010). Only pixels lying within the rectangular grid are preserved, the rest of the pixels are disposed.

An image has 100×100 pixels. Downsampling factor of 100 is chosen, $c = 100$. It is assumed that downsampling factor to be a square, so that its width equals its height.

$$c_{width} = c_{height} = \sqrt{c} = \sqrt{100} \approx 10 \quad (3.6)$$

Hence, the resultant image after downsampling has width, w' and height, h' .

$$w' = h' = \frac{100}{c_{width}} = \frac{100}{10} \approx 10 \quad (3.7)$$

The total features after downsampling is as follows:

$$w' \times h' \times \text{number of filter} = 10 \times 10 \times 40 = 4000 \text{ features.} \quad (3.8)$$

Thus, 400,000 features are streamlined to 4000 features.

3.4.2 Oriented Gabor Phase Congruency Filters

Gabor magnitude changes slowly across the spatial location, but Gabor phase information can have very different values even if it is measured a few pixels apart. According to Zhang et al. (2007), the face representations from phase are determined rather than the raw phase responses are used as face descriptors. According to Kovesi (2000), he employed the phase congruency model to look for points within an image where the log-Gabor filter outputs are maximally in phase with each other. The phase congruency is computed for every filter orientations and is combined to obtain a final phase congruency output image. His work claimed that phase congruency output robustly detects the edges and corners. However, doubts are posed by Liu in (Liu, 2006) that Kovesi's method might not be capable of extracting features of multi-orientations effectively.

For the implementation of feature extraction in this project, Gabor phase congruency information for multi-orientations are computed and then augmentation of the phase congruency feature vectors are carried out. Instead of log-Gabor function used by Kovesi, conventional Gabor function is used for filters with 5 scales and 8 orientations. The definition of OGPC is defined as follows (Struc et al, 2009):

$$OGPC = \frac{\sum_{u=0}^{p-1} A_{u,v}(x, y) \Delta \phi_{u,v}(x, y)}{\sum_{u=0}^{p-1} (A_{u,v}(x, y) + \epsilon)} \quad (3.9)$$

Where ϵ is a constant to prevent equation divided by zero; $\phi_{u,v}(x, y)$ is the phase angle of Gabor function; u is the scales and v is the orientations; p is the total number of scales;

$\Delta\phi_{u,v}(x, y)$ is the phase deviation given by:

$$\begin{aligned} & \Delta\phi_{u,v}(x, y) & (3.10) \\ & = \cos\left(\phi_{u,v}(x, y) - \bar{\phi}_{u,v}(x, y)\right) - \left|\sin\left(\phi_{u,v}(x, y) - \bar{\phi}_{u,v}(x, y)\right)\right| \end{aligned}$$

Where $\phi_{u,v}(x, y)$ is the phase angle of Gabor filter response; $\bar{\phi}_{u,v}(x, y)$ is the average phase angle of Gabor filter response.

The output of OGPC is the illumination and contrast independent facial feature representations. The OGPCs are then downsized by downsampling factor. Z-score normalization is applied to the downsized OGPCs and the OGPC feature column vectors are concatenated together becoming augmented OGPC feature vectors.

For OGPC, eight orientations information is generated for every single pixel intensity of an image. Thus for an image of size 100×100 pixels, the total number of features would be 80,000 ($=100 \times 100 \times 8$). The factor of scale magnitude is negligible in the Gabor phase feature generation. Thus the number of features generated due to Gabor phase components is 5 times lesser than Gabor magnitude's. Due to the large dimensionality of 80,000 features causing the exorbitant computational costs, the downsampling factor value of 100 reduces the feature number to 800 ($=80,000/100$).

3.4.3 Maximum Response Filters

Maximum Response (MR) filters has 38 filters and 8 filter responses. The filters include one Gaussian and one Laplacian of Gaussian isotropic filters at scale $\sigma = 10$. The Laplacian of Gaussian isotropic filters detects the features with greater discrimination power. There are also edge (first derivative) anisotropic filters with 6 orientations and 3 scales and bar (second derivative) anisotropic filters with 6 orientations and 3 scales with $(\sigma_x, \sigma_y) = \{(1,3), (2,6), (4,12)\}$, according to

Geusebroek et al. (2003). The isotropic filter responses are utilized without any further processing; for the anisotropic filter only the maximum filter responses at every scale across entire orientations are chosen. This generates 8 filter responses and they are rotational invariant. The respective equations are shown as follows (Geusebroek et al., 2003):

$$\text{Edge filter} = -\exp\left[-\frac{\text{input}^2}{2\sigma^2} \cdot \frac{1}{\sqrt{2\pi\sigma^2}}\right] \left(\frac{\text{input}}{\sigma^2}\right) \quad (3.11)$$

$$\text{Bar filter} = \exp\left[-\frac{\text{input}^2}{2\sigma^2} \cdot \frac{1}{\sqrt{2\pi\sigma^2}}\right] \cdot \left(\frac{\text{input}^2 - \sigma^2}{\sigma^4}\right) \quad (3.12)$$

$$\text{input} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} * (\text{pixel value of image}) \quad (3.13)$$

where $\theta = \frac{n\pi}{6}$, $n = 0, 1, 2, \dots, 5$ and σ is the standard deviation along x-axis and y-axis.

Similar to Gabor filter, the facial images are convolved with the MR filter banks to produce the output response. Figure 3.6 below shows the filter response at different orientations and scales of the anisotropic and isotropic filters. The reason of using Maximum Response filters is that they have bar filters.

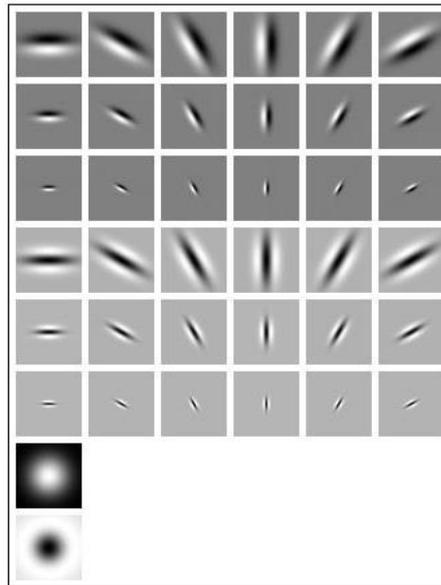


Figure 3.6: The filter output of the MR filters. First three rows are edge filters. The fourth until sixth rows are bar filters. The remaining two filters are isotropic Gaussian and Laplacian of Gaussian filters

3.5 Feature Selection

The details about the purpose and several types of feature selection have been explained in Chapter 2. In this project, Monte Carlo Uninformative Variable Elimination partial least square regression is used in feature selection process due to its lesser time-consuming compared to genetic algorithm feature selection.

3.5.1 Monte Carlo Uninformative Variable Elimination Partial Least Squares Regression (MC-UVE-PLSR)

The combination of Monte Carlo (MC) method and Uninformative Variable Elimination (UVE) method is used to select features generated by Gabor filters. Each feature's set reliability is evaluated based on its stability. Usually the UVE employs the leave-one-out procedure, however Monte Carlo method is used instead of the leave-one-out procedure (Cai, 2008).

The samples are divided randomly into training set, evaluation set and prediction set. The Monte Carlo randomly choose a number of subsamples from

training set (at 75%) to build the partial least square (PLS) model and this process repeats 1000 times. The PLS regression coefficients and the stability of each feature set are computed.

$$\gamma = \alpha X + \kappa \quad (3.14)$$

Where γ is the prediction; X is the information of the feature sets; α is the regression coefficients; κ is the offset.

The regression coefficients, α_i define the contribution of that particular feature to the prediction model. The reliability of a feature is evaluated using its stability level, s_i :

$$s_i = \frac{\text{mean}(\alpha_i)}{\text{standard deviation}(\alpha_i)} \text{ where } i = 1, 2, \dots, \text{ number of feature} \quad (3.15)$$

The higher level of stability means that particular feature is more important. The features are rank based on the stability level from the highest to the lowest. The higher the reliability ranking of a feature, the more important it becomes. From the features generated by Gabor and Maximum response filters, the top ranked 2000 features are selected to be fed into Random Forest classifier function. The reason of selecting 2000 features is that it is the balance point between maintaining optimal accuracy level and the computation costs in implementing the training process (will be discussed in Chapter 4). For OGPC, the feature selection is not to be applied onto it due to its already optimal dimension size at 800 features.

3.6 Classification and Prediction Using Learning Framework

The classification is used to identify the category that a new observation is belonged to. During training phase, the training dataset has the information of observations

and their categorical memberships. In this project, the Random Forest algorithm is used for classification.

3.6.1 Random Forests

The random forest can handle high dimensional spaces and large number of training samples. Each node in random forest is split using randomly selected features. According to Ho (1998), the collection of random set of features is able to tackle the challenge of data overfitting.

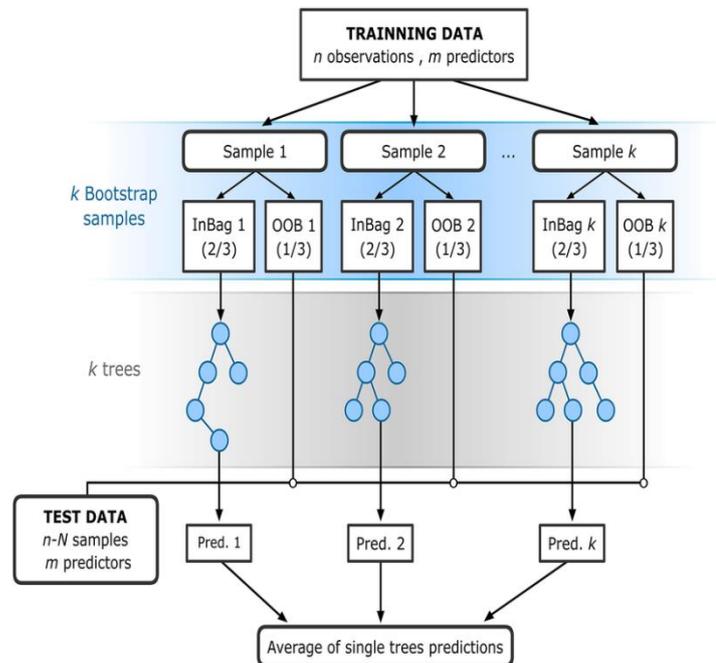


Figure 3.7: Flowchart of the random forest (Rodriguez-Galiano et al., 2015)

Bootstrapping is used in random forest for training or testing. Bootstrapping refers to the generation of sub-samples randomly from the provided dataset (at two-third of total number of samples) but it is with replacement from the initial data (Breiman, 2001). The remaining one-third are used to make prediction against the bootstrapped samples to estimate the error of the tree model. When a number of randomized bootstrapped samples are generated and the decision trees are grown, for every occasion a different prediction will be obtained. The final prediction would be

the mean of all 100 predictions. Random Forest differs from boosting in the aspect of the weight of each prediction. In boosting, lower weights are assigned to those predict less accurately and higher weights are assigned to those predict more accurately. Thus in boosting, final result is the weighted average of all predictions. In Random Forest however, all predictions are assigned with same weight.

Random forest is built using N classification stages, for which N is the number of trees. The values of N used to investigate the error rate of the Random Forest model are 100, 200, 300, 400 and 500 (to be discussed in Chapter 4). During the process of classifying a test sample, input feature vectors of test image are assessed on each tree of the random forest. Each tree generates a classification result which is a “vote”. The class that has the maximum number of votes is declared as the final classification result.

The tree is grown in this way:

1. The M number of training sample and P number of features are used.
2. In-bag sample is chosen for k times with replacement from M training samples (two-thirds of M). The remaining one-third is the out-of-bag sample and it is used to estimate the error of tree.
3. At the tree decision node, J number of features is selected randomly from P . The best split decision among J variables is calculated.
4. The tree is grown to largest size possible without pruning.
5. N number of trees are grown during the training stage and the outputs of the decision trees are averaged and the class that gets highest votes is declared as final prediction.

The tree is expanded using in-bag sample, which is two-third of training sample. The remaining one-third is excluded. The remaining sample is the out-of-bag samples (OOB) which can be used to get the value of J . According to Breiman (2001), the random forest’s rate of error is dependent on the following factors:

1. The correlation between two trees in the random forest. The higher correlation translates into higher forest's rate of error.
2. Strength of the tree. The greater strength translates into lower random forest error rate.

When the value J is increased, the correlation and strength are increased as well. The optimum value of J is required to balance between these two contradicting outcomes. OOB error is used to calibrate the J to get the optimal value. OOB sample is processed throughout the completely formed tree to reach the classification output. OOB error is computed through the number of wrongly classified samples, and is averaged over the entire cases. J value is attuned in order to minimize the OOB error. For starting, J is initialized at the value equivalent to the square root of the total numbers of features ($J = \sqrt{P}$). The J is then continually attuned to an optimal value which corresponds to minimum OOB error.

For instance, there are 2000 possible features for selection. At the beginning, the random selection of 44 features ($44 \approx \sqrt{2000}$) is obtained and best split decision is selected. The root node is divided into left and right nodes by comparing input feature value and the threshold value. If the input feature value is less than the threshold value, the tree will advance to the left node. Otherwise, it will advance to right node. If the node reaches at only one class, the tree stops expanding after that particular node. The terminal node is called leaf node. The splitting continues until the leaf node is reached. The output of leaf node is a class. The ensemble of random decision trees makes up the random forest. The reason of growing the decision tree to maximum size without trimming is that bias can be maintained at low level and overfitting problem can be evaded. According to Breiman (2001), the higher number of trees converges the generalization error is small. Breiman (2001) claimed that 500 trees is an optimal number of trees to deliver a reliable result.

During the testing stage, the input test image undergoes the image pre-processing and feature extraction processes. The feature vectors of the input test image will traverse the decision trees in the trained model. For example the upper

parent node shown in Figure 3.8 generates the split criterion at 0.825084. The split criteria is based on Gini criterion which is shown as follows (Li, 2010):

$$\text{Gini} = n_L \sum_{k=1}^K r_L(1 - r_L) + n_R \sum_{k=1}^K r_R(1 - r_R) \quad (3.16)$$

Where n_L is the node on the left; n_R is the node on the right; r_L is the proportion of class k in left node; r_R is the proportion of class k in right node; K is total number of classes.

The purpose of split criterion is to decrease the Gini impurity value of child node, ie. higher proportion of features reaching at child nodes are belonged to the same class than the proportion at parent node. In other words, features at child nodes are “purer” than at parent nodes in terms of their probable class membership.

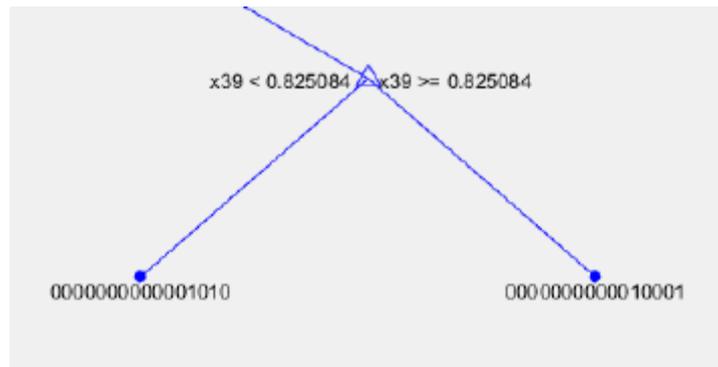


Figure 3.8: Visualization of decision split

3.7 Fusion of Gabor Filters and Maximum Response Filters

According to Vitomir (2010), the fusion between magnitude and phase information of Gabor filter is believed to be able to build a robust facial feature extraction. The effectiveness of the feature extraction method is measured by the face recognition rate.

By combining matching score of the classifier output of the Gabor magnitude filters, $S_{magnitude}$ and Oriented Gabor Phase Congruency Image, S_{phase} , a final matching score of complete Gabor Classifier S_{gabor} is computed as follows:

$$S_{gabor} = (1 - r) * S_{magnitude} + (r) * S_{phase} \quad (3.17)$$

Where r sets the proportion between Gabor magnitude and phase components and $r \in [0, 1]$.

In this project, the fusion between Gabor filter and Maximum Response Filter is further carried out to enhance the feature extraction capability and improve recognition rate as shown in Figure 3.9. Similar to the fusion between magnitude and phase information of Gabor filter, the matching score of the classifier output of the Gabor filters, S_{gabor} and Maximum Response filters, S_{MR} , a final matching score of complete Gabor-MR fusion Classifier $S_{gabor-MR}$ is computed as follow:

$$S_{gabor} = (1 - q) * S_{magnitude} + (q) * S_{phase} \quad (3.18)$$

Where q sets the proportion between Gabor and Maximum Response components and $r \in [0, 1]$.

The value of q is further discussed in Chapter 4.

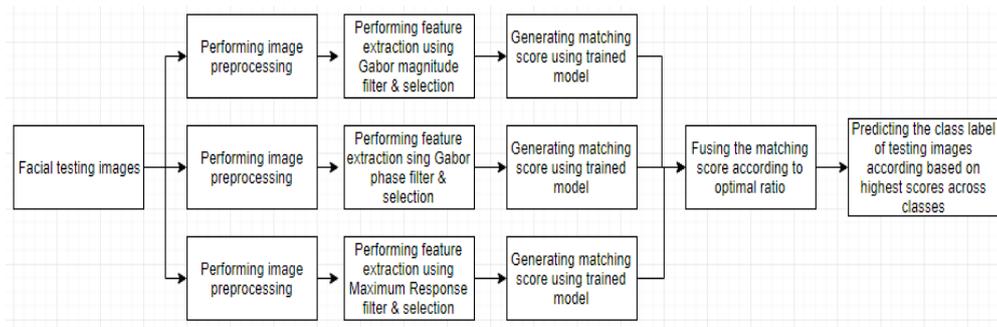


Figure 3.9: Fusion between Gabor and MR filters

3.8 Software

The program is written in MATLAB 2014b. The MATLAB contains built-in image processing toolbox and statistical toolbox which simplify the code writing process. The feature extraction library called PhD toolbox developed by Vitomir Struc is used as it contains algorithms such as principal component analysis, linear discriminant analysis and Gabor filter functions. MATLAB has also a library for random forest function. It is called TreeBagger which enables the user to predict the scores for each class and view the decision tree of the classification.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

The recognition rate of the system is defined as follows:

$$\text{Recognition rate} = \frac{\text{number of correct classification}}{\text{number of correct classification} + \text{number of wrong classification}} \quad (4.1)$$

The greater the recognition rate, the higher the number of face photos are being identified correctly. The recognition rate is the biometric evaluation tool to evaluate the performance of the algorithm.

4.2 The Pre-Processing Phase

4.2.1 Face Detection

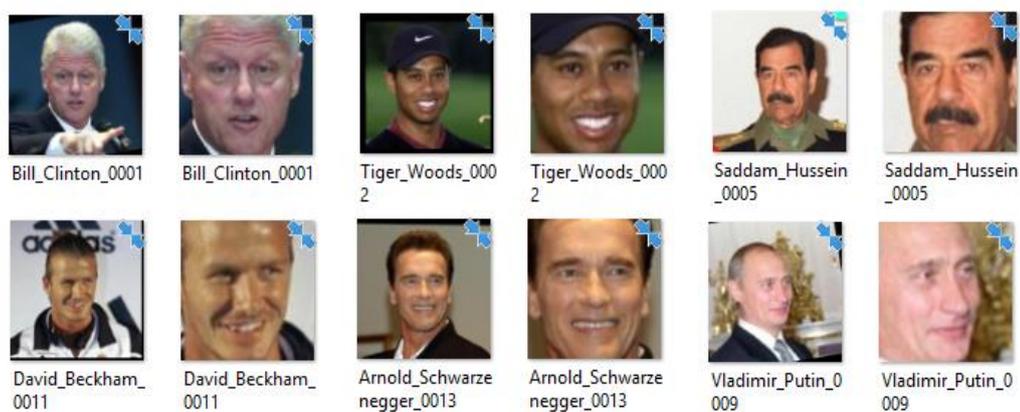


Figure 4.1: The output of the face detection after resizing

The face detection using the Viola-Jones face detector is carried out and the facial regions of interest are cropped out and are saved separately from the original image files. The cropped regions of the facial images are resized to 100×100 pixels for consistency. This is shown in Figure 4.1.

4.2.2 Conversion into Grayscaled Format

In the preprocessing phase, each of the training images or testing images is converted into grayscale format from RGB color format. The grayscale format reduces the dimensionality of the color information of an image thus reduce the computation complexity and costs. This is shown in Figure 4.2.



Figure 4.2: Conversion into grayscale image

4.2.3 Tan-Triggs Illumination Normalization

As mentioned in Chapter 3, Tan Triggs illumination normalization is used to suppress the extremities of intensity values of the image. This is to reduce the masking of the useful feature information due to shading. Thus, the entire face features information are preserved. This is shown in Figure 4.3.





Figure 4.3: Effect of Illumination normalization to counter the influence of shading or overexposure of light

Table 4.1: Study of the effect of Tan-Triggs illumination normalization

Database	LFW (without Tan-Triggs Illumination Normalization)	LFW (with Tan-Triggs Illumination Normalization)
Ni, η	0.9428	0.9428
Gamma, γ	sqrt(2)	sqrt(2)
Maximum central frequency, F_{\max}	sqrt(2)/4	sqrt(2)/4
Correct classification	392	422
Wrong classification	201	171
Recognition rate	0.6610	0.7116
ROC	0.9452	0.9540

As shown in Table 4.2, if Tan-Triggs illumination normalization is used, the recognition rate rises from 0.661 to 0.7116. This can be explained by the effect of the variation of illumination levels among the samples on the feature extraction process. The feature extraction process takes into account of the gradient of the pixel intensities and the orientation changes of the intensities. The illumination

normalization evenly subdues the extreme pixel intensity level to counter the shading or flashing effects of the facial images.

4.3 Feature Extraction

In this stage, there are three sub-stages, which are Gabor magnitude image filtering, oriented Gabor phase congruency (OGPC) image filtering and Maximum Response image filtering. These image filterings are applied on the pre-processed images. The features generated by Gabor magnitude, OGPC and MR filters will be fed into Random Forest classifiers during the training process to construct the decision trees.

4.3.1 Gabor Filters

For Gabor filters, there are 8 orientations and 5 scales of magnitude of filters which make up to 40 filters in total. For a 100×100 pixels of image, there are 40 features convolved for each pixel. Thus the total number of features generated by the Gabor magnitude filter for a 100×100 pixels image are $100 \times 100 \times 40 = 400\,000$ features per image.

The overall image downsampling factor is set at 100 after trials-and-error. By square-rooting the overall image downsampling factor we will get the image-side-length downsampling factor at $\sqrt{100} = 10$.

The downsampling factor of image-side-length is the new image side dimension after image resizing. Thus the number of features will be reduced to $\frac{400000}{10 \times 10} = 4000$ features per image.

The reduced number of features reduces the computation complexities and costs during the classification process in the later stages. The outputs of Gabor magnitude filter banks is shown in Figure 4.4.

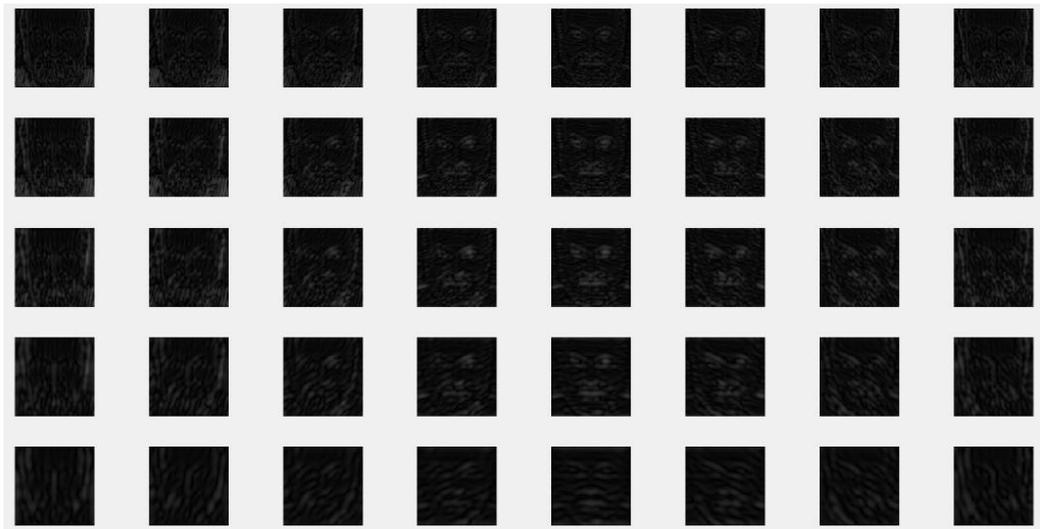


Figure 4.4: Gabor magnitude response

For OGPC, there are 8 orientations of the image filter. Thus filter convolution of each pixel of the image generates 8 features per pixel. The total number of features generated is $100 \times 100 \times 8 = 80\,000$ features per image. The downsampling factor at 100 reduces the number features to $\frac{80000}{100} = 800$ features per image.

The output of the OGPC filter banks is shown as follows:

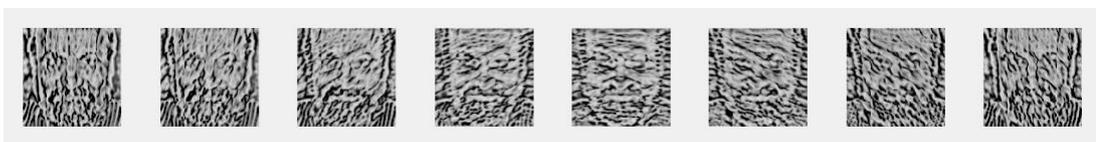


Figure 4.5: Gabor Phase Response

4.3.1.1 The Effect of Gabor Magnitude Filter Parameters on Recognition Rate

Table 4.2: Study of the effect of tuning the Gabor filter parameter on the recognition rate

Column	1 (Turner, 1986)	2 (Vitomir, 2010)	3 (proposed in this project)
Database	LFW	LFW	LFW
Ni, η	0.942809041	sqrt(2)	0.5
Gamma, γ	sqrt(2)	sqrt(2)	0.5
Maximum central frequency, F_{max}	sqrt(2)/4	0.25	sqrt(2)/4
Correct classification	392	390	407
Wrong classification	201	203	186
Recognition rate	0.661	0.6577	0.6863

The general equation of the 2D-Gabor filter is shown below:

$$Gabor\ filter = \frac{F_{max}^2}{\pi\gamma\eta} e^{-F_{max}^2[(\frac{x'}{\gamma})^2 + (\frac{y'}{\eta})^2]} e^{i2\pi Fx'} \quad (4.2)$$

$$x' = x\cos\theta + y\sin\theta$$

$$y' = -x\sin\theta + y\cos\theta$$

Where F_{max} is the maximum frequency of filter; θ is the angle between sinusoidal wave direction and the x-axis in spatial space; γ and η are the standard deviations of Gaussian function parallel to and perpendicular to the wave direction and they define the selectivity of filter in spatial space.

The values of parameters (η , γ and F_{max}) of Column 1 of Table 4.2 are recommended by Turner with recognition rate rises of 0.661.

The values of parameters of Column 2 of Table 4.2 are recommended by Vitomir with recognition rate rises of 0.6577. However, with the proposed parameters, the recognition rate rises to 0.6863.

Gamma, γ or the spatial aspect ratio determines the ellipticity of Gaussian function and the width of the Gaussian window. N_i , η specifies the linear size of the visual receptive field simulated by the Gabor filter. It is found that by setting to lower values of Gamma and N_i , the finer discrimination of the texture of the facial region is obtained. Based on empirical result as shown in Table 4.2, the higher value of the maximum central frequency also improves the recognition rate. According to Xiao (2007), for the frequency-transformed even symmetric Gabor filter, the resultant transfer function is the addition of two Gaussian functions centered at positive and negative central frequencies shown in Figure 4.6.

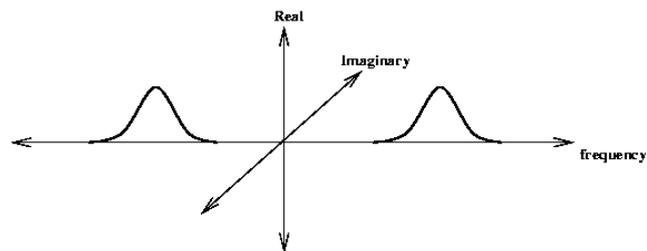


Figure 4.6: Even symmetric filter transfer function

The lower Gamma value gives smaller Gaussian bandwidth and sharper filter so that the tails of the two Gaussians do not overlap much at the origin which produces only a few non-zero DC components. The lower value of Gamma is crucial in obtaining maximal spatial localization of frequency information of the facial image.

The higher value of maximum central frequency, F_{max} moves the two Gaussian functions further apart so that the overlap does not happen excessively. This is to restrain the frequency value within the scope of Nyquist frequency. The

excessive overlapping and higher frequency bandwidth (due to higher Gamma) will cause smaller coverage of the spectrum in spatial domain. Since 40 Gabor filters are used, the excessive overlapping will cause narrower spectrum of features being detected and extracted, thus lowering down the recognition rate (Peterkovesi.com, 2017).

Therefore, the parameters (η , γ and F_{max}) of the Column 3 of Table 4.2 are adopted in the subsequent benchmark tests to investigate the recognition accuracy.

4.3.2 Maximum Response (MR) Filters

Unlike Gabor filters, Maximum Response (MR) filters have two types of anisotropic filters (6 orientations and 3 scales for each type) and two isotropic filters, making up 38 filters in total. For a 100×100 pixels of image, there are 38 feature outputs generated for each pixel. Thus the total number of features generated by the Gabor magnitude filter for a 100×100 pixels image are $100 \times 100 \times 38 = 380\,000$ features per image.

The downsampling factor is set at 100. By square-rooting the downsampling factor the downsampling factor of image-side-length is $\sqrt{100} = 10$.

The downsampling factor of image-side-length is the new image side-length after image resizing. Thus the number of features will be reduced to $\frac{380000}{10 \times 10} = 3800$ features per image.

Similar to the scenario in Gabor feature extraction, the reduced number of features reduces the computation complexities and costs during the classification process in the later stages. The output of Maximum Response filter banks is shown as follows:



Figure 4.7: MR Filter Response

4.4 Feature Selection

The amount of features generated by Gabor filters and MR filters is still large and it causes heavy computation costs during the classification process. Thus, 2000 most useful features are selected through the Monte Carlo Uninformative Variable Elimination PLS Regression method (*MC-UVE-PLSR*). The selection of the 2000 highest importance features is decided to strike an optimal balance between maintaining optimal classification accuracy level and controlling the computation costs incurred during classification process.

Table 4.3: Study of the effect number of features and feature selection on the recognition rates

Column	1	2	3	4	5
Database	LFW	LFW	LFW	LFW	LFW
Ni, η	0.9428	0.9428	0.9428	0.9428	0.9428
Gamma, γ	sqrt(2)	sqrt(2)	sqrt(2)	sqrt(2)	sqrt(2)
Maximum central frequency, F_{max}	sqrt(2)/4	sqrt(2)/4	sqrt(2)/4	sqrt(2)/4	sqrt(2)/4
Resolution	100×100	100×100	100×100	100×100	100×100
Downsampling factor	400	200	100	50	100
Feature Selection	No	No	Yes	Yes	No
Number of features generated before feature selection process	1000	2000	4000	8000	4000
Number of features to be	1000	2000	2000	2000	4000

selected					
Feature extraction time (seconds)	817.2631	713.7431	1530.3125	5656.3488	321.1947
Classification time (seconds)	4310.0063	7770.5528	8201.3336	7579.6424	23475.5223
Correct classification	315	427	443	435	468
Wrong classification	278	166	150	158	125
Recognition rate	0.6610	0.7201	0.7470	0.7336	0.7892

By comparing the Column 1 and Column 2 of Table 4.3, it is found that the higher the number of features used during Random Forest classification process, the recognition rates are improved significantly from 0.6610 to 0.7201. By comparing the Column 2 and Column 3, the difference in downsampling factor generates different number of features. By selecting the same number of features (at 2000 highest importance features) to be fed into the classifier, it is found that the 2000 features chosen from the larger pool of 4000 features will register higher recognition rates than selecting all features from the smaller pool of 2000 features. This is because the smaller downsampling factor, the greater the number of facial feature information are retained.

By inspecting Column 3 and Column 4, there is a limit of the downsampling factor in achieving the optimal recognition rate. The recognition rate using 2000 features selected from the pool of 8000 features is lower than selecting the same feature amount from the pool of 4000 features. This is because each feature of the pool of 8000 features describes the local feature with greater localization and narrower frequency scope. Therefore, the higher the number of features are selected from the pool of 8000 features in order to be more globally representative in describing a facial image.

The *MC-UVE-PLSR* process ranks the features according to their information importance (explained in Chapter 3). A subset of features with the highest information importance are selected. The reason of using feature selection is explained as follows. If the number of features fed into the classifier is too high, the classification process consumes disproportionately longer time and it is unfeasible in terms of limited computation resources. Based on the Column 3, it is found that the recognition rate is improved by implementing the feature selection process out of the larger pool of features.

By comparing Figure 4.8 and Figure 4.9, it is found that the features with the higher feature importance are more compact and denser in the smaller pool of 2000 features (Figure 4.8) compared with that in the larger pool of 4000 features (Figure 4.9). This shows that the feature selection using *MC-UVE-PLSR* helps to improve computation costs.

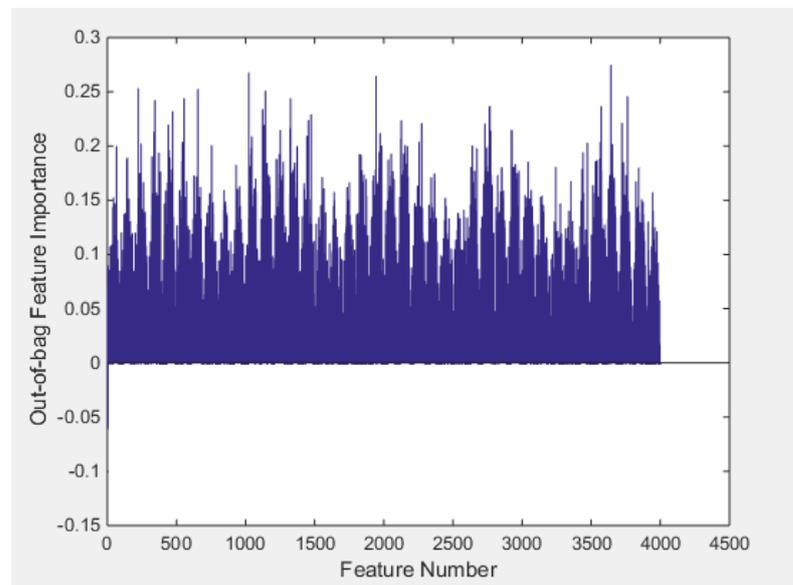


Figure 4.8 : The positive-valued OOB Feature Importance among the pools of 4000 features

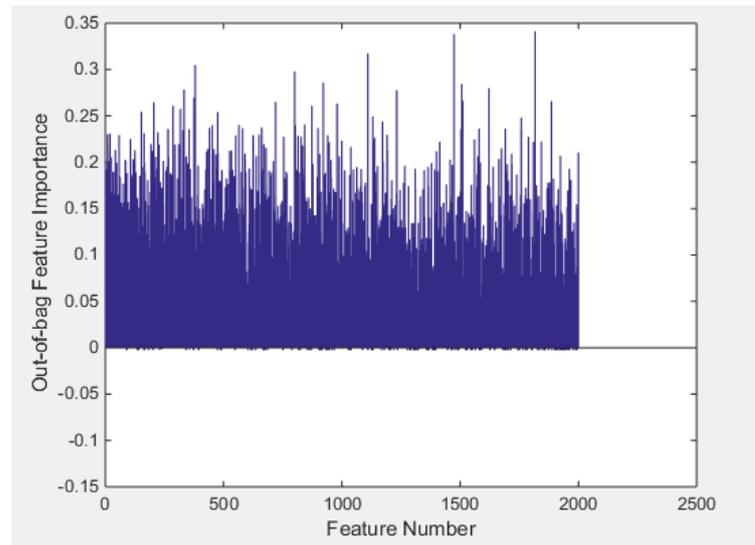


Figure 4.9: The positive-valued OOB Feature Importance among the pools of 2000 features

By comparing Column 3 and Column 5, although the full selection of the 4000 features without the feature selection process (Column 4) obtains a higher recognition rate, the classification process in Column 4 consumes more time than that in Column 3. The computation costs are high and are not suitable without the feature selection.

4.5 Classification using Random Forests

The MATLAB's built-in `Treebagger` function (Mathworks.com, 2017) is used in classification process to create an ensemble of decision trees to reduce the problem of overfitting. The bootstrapping is performed in this stage. Every tree is grown independently by chosen bootstrapped sub-samples of the input dataset. The bootstrapping randomly selects two-third of samples from input data set, the remaining one-third are used as out-of-bag sub-samples to make prediction on the bootstrapped sub-samples.

The decision trees are not pruned and are allowed to grow into maximum depth size. The weights of growing every decision tree are all set to 1. This implies that each tree has the same weightage and each tree has the same level of influence during classification voting. The minimum number of observation for every tree leaf

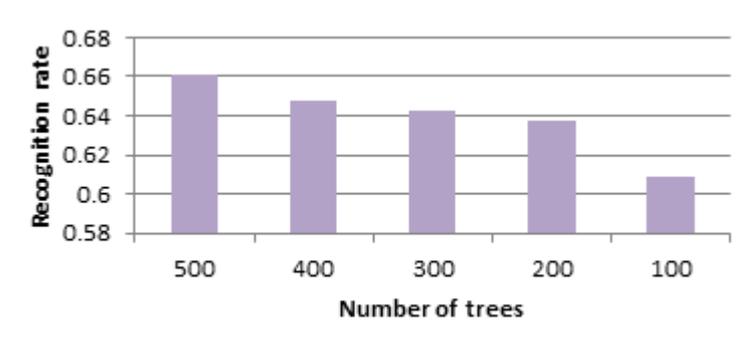


Figure 4.11: Effect of number of decision trees on Recognition rate

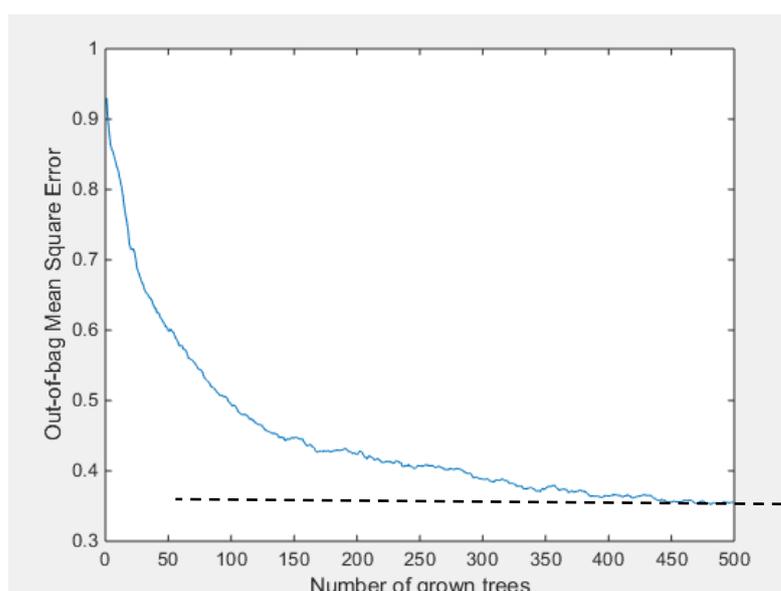


Figure 4.12: Effect of number of grown decision trees on Out-of-bag mean square error

Based on Figure 4.11 and Figure 4.12, it is found that the greater the number of trees used in Random Forest, the lower the out-of-bag mean square error, the higher the recognition rates of the system. The out-of-bag-error (OOB Error) starts to plateau in the region between 450 and 500 trees. The higher number of grown trees provides higher total classification votes so the deviation of the output of one tree does not deviate the mean of the voting result by too large margin. This proves that number of trees at 500 is optimal to minimize the mean square error and to improve the classification accuracy.

4.6 Fusion of the Random Forest Prediction Scores of Gabor Magnitude, OGPC and MR filters

The scores produced by each decision tree are the probability of the sample belonging to a particular category. The mean score averaged over all decision trees in the ensemble is used as the final score output of classifier.

It is found that the fusion of the Gabor filters and MR filters increases the recognition rate of the recognition system in comparison with the Gabor-only feature extraction. The fusion prediction scores are computed as follows:

Gabor-only prediction scores

$$= (1 - y) \times \text{Gabor magnitude scores} + (y) \times \text{OGPC scores}$$

Fusion prediction scores

$$= (1 - r) \times \text{Gabor only prediction scores} + (r) \times \text{MR only prediction scores}$$

, where y and r are the ratio parameters of the Gabor-only prediction scores and fusion prediction scores.

The total prediction scores of each of the testing images are obtained by fusing the scores of Gabor-magnitude, OGPC and MR filters by a ratio. The predicted class is determined by the class which gets the highest scores. Based on Figure 4.13 and Figure 4.14, the highest recognition rate at 0.8128 is achieved at Gabor phase-magnitude ratio (0.48) and MR-Gabor ratio (0.35).

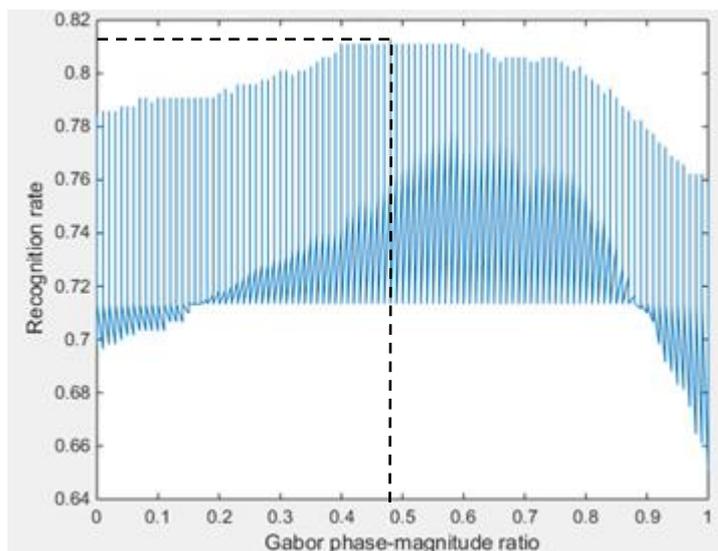


Figure 4.13: Effect of Gabor phase to Gabor magnitude ratio on Recognition rate

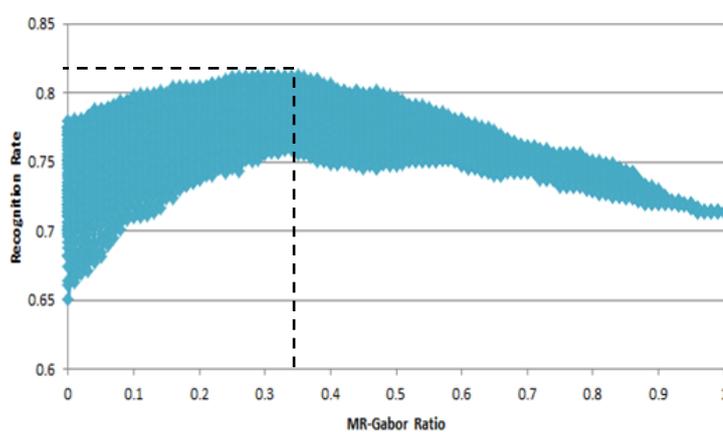


Figure 4.14: The graph of Recognition rate against the MR-Gabor ratio

Based on the obtained results as shown in Figure 4.14 and Figure 4.15, the fusion among these filters improves the recognition rate. The Gabor phase-magnitude ratio (0.48) and MR-Gabor ratio (0.35) are used to test the reliability of the LFW database.

4.7 Finalization of Parameters

The summarized parameters are set in Tables 4.4 and 4.5 for the benchmark tests in Chapter 4.8.

Table 4.4 : Summarized parameter settings of LFW database

Database	Labelled faces in the wild (LFW)
Face frontalization	No
Illumination Normalization	Yes
Resolution after resize (pixels)	100 × 100
Ni, η	0.5
Gamma, γ	0.5
Separation	Sqrt(2)
Fmax	Sqrt(2)/4
Orientation	8
Scale	5
Number of trees	500
Number of features generated before feature selection process	4000
Number of features to be selected	2000
Feature extraction time (seconds)	2964.379146
Classification time (seconds)	8507.332345
Gabor phase to magnitude ratio	0.48
MR to Gabor ratio	0.35

Table 4.5: Summarized parameter settings of UFI database

Database	Unconstrained Facial Images (UFI)
Face frontalization	No
Illumination Normalization	Yes
Resolution after resize (pixels)	100 × 100
Ni, η	0.5
Gamma, γ	0.5
Separation	Sqrt(2)
Fmax	Sqrt(2)/4
Orientation	8
Scale	5
Number of trees	500
Number of features generated before feature selection process	4000
Number of features to be selected	2000
Feature extraction time (seconds)	1325.033278
Classification time (seconds)	11410.563557
Gabor phase to magnitude ratio	0.48
MR to Gabor ratio	0.35

4.8 Comparison with Other Existing Algorithms

4.8.1 The Use of Cross Validation in the Evaluation of Algorithms

Cross validation is used in DET, ROC and Recognition rate measurements to evaluate how the outcome of the classification analysis generalizes to an independent dataset. In other words, it is the estimation of the performance of the predictive model. In cross validation, the testing dataset is used to test the performance of model which can minimize the overfitting problem.

The cross validation process divides the data samples into several subsets. One of the subsets will be the testing dataset and the remaining subsets will be the training datasets. Several rounds of cross validation are carried out using different subsets and the validation outcome is the mean value over the number of rounds.

	Round 1	Round 2	Round 3	Round 4	Round 5
Data Subset 1	<i>Test</i>	Training	Training	Training	Training
Data Subset 2	Training	<i>Test</i>	Training	Training	Training
Data Subset 3	Training	Training	<i>Test</i>	Training	Training
Data Subset 4	Training	Training	Training	<i>Test</i>	Training
Data Subset 5	Training	Training	Training	Training	<i>Test</i>

Figure 4.15: Cross validation

Cross validation is a more accurate demonstration of the performance of the predictive model. 5-fold cross validation is used in this project. The data samples are randomly divided into 5-equal sized subsets. One subset is used as the testing dataset and the remaining 4 subsets are used as training datasets. The cross validation will be carried out 5-times in one iteration. Each subset is used as testing dataset exactly once throughout the 5-rounds of cross validation. The results are then averaged out to obtain the final estimation. The advantage of this method is that the entire samples are used for both training and testing while each sample is only used for testing for exactly once. The visualization of cross validation is shown in Figure 4.15.

4.8.2 Labelled Faces in the Wild

4.8.2.1 Confusion Matrix

Confusion matrix is an error matrix. It visualizes the performance of the algorithm. The columns of the matrix are the predicted class. The rows of the matrix are the actual class.

Confusion matrix is used to observe how the classifier algorithm is confused between classes and how well it distinguishes between classes accurately. Each diagonal cell of the matrix represents the correct classification. The value of each of the cell represents the percentage of the predicted class is correctly matched onto the actual class. The closer the value of all diagonal cells to 100%, the higher the number of correct classification so the better the algorithm performs. The confusion matrix of implemented method applied on LFW database is shown in Figure 4.16 and 4.17.

CLASS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	0.888889	0	0	0	0	0	0	0	0	0.111111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0.666667	0	0	0	0	0	0.166667	0.166667	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0.966667	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0.083333	0.583333	0	0	0	0	0	0	0	0	0.083333	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0.75	0	0	0	0	0	0	0	0.25	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0.111111	0.111111	0	0	0.222222	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.111111	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0.933333	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0.033333	0	0	0	0	0	0.933333	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0.033333	0	0	0	0	0	0	0	0.033333	0.033333	0	0	0.8	0.066667	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0.033333	0	0	0	0	0.033333	0.033333	0	0	0	0.766667	0	0	0	0	0	0	0	0.033333	0.033333	0	0	0	0
16	0	0	0	0	0	0.071429	0	0	0	0	0	0	0	0	0	0.857143	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0.166667	0	0	0	0	0	0	0	0	0.166667	0	0.666667	0	0	0	0	0	0	0	0	0	0	0	0
18	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
21	0.033333	0	0	0	0	0	0	0	0	0	0	0	0	0	0.033333	0	0	0	0	0	0	0.733333	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.92	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0.083333	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.416667	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5

Figure 4.16: Confusion Matrix of Gabor-MR-fusion with Random Forest on LFW database

Class	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57			
29	0.826087	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.086957	0	0	0	0	0	0	0.043478	0	0	0	0	0		
30	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
31	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
32	0	0	0	0.75	0	0	0.125	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.125	0		
33	0	0	0	0	0.933333	0	0	0	0	0	0	0	0	0	0	0	0	0.033333	0	0	0	0	0	0	0	0	0	0	0	0		
34	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
35	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
36	0.5	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
37	0	0	0	0	0	0	0	0	0.727273	0	0	0	0	0	0	0	0	0.090909	0	0	0	0	0	0	0	0	0	0	0	0	0	
38	0	0	0	0	0	0	0	0	0	0.833333	0	0	0	0	0	0	0	0.055556	0	0	0	0	0	0	0	0	0	0	0	0	0	
39	0	0	0	0	0	0	0	0	0	0	0.777778	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
40	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
44	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.857143	0	0	0	0	0	0	0	0	0	0	0	0	0	
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.166667	0	0	0	0	0.166667	0	0	0	0	0	
48	0	0	0	0	0	0	0.333333	0	0	0	0	0	0	0	0	0	0	0	0	0	0.666667	0	0	0	0	0	0	0	0	0	0	
49	0	0	0	0	0	0	0.090909	0	0.045455	0	0	0	0	0	0	0	0	0	0	0	0	0.863636	0	0	0	0	0	0	0	0	0	
50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
51	0	0	0	0.333333	0	0	0	0	0	0	0	0	0	0.333333	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
52	0	0.2	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
54	0.033333	0.066667	0	0	0.033333	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0.433333	0	0.066667	0	0	
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
56	0	0.052632	0	0	0	0	0	0	0.052632	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.736842	0	0
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Figure 4.17: Confusion Matrix of Gabor-MR-fusion with Random Forest on LFW database (continued)

4.8.2.2 Recognition Rate or Classification Accuracy

Table 4.6: Comparison of Classification Accuracy among Different Algorithms on LFW Databases (Learned-Miller, 2016)

Algorithm	Classification Accuracy (Highest = 1.0)
Gabor-MR-Fusion-Random Forest	0.8128
V1-like/MKL, funnelled (Xu et al., 2015)	0.7935
MRF-MLBP	0.7908
Hybrid descriptor-based, funnelled (Xu et al, 2015)	0.7847
Nowak, funneled	0.7393
3x3 Multi-Region Histograms (1024)	0.7295
Nowak, original	0.7245
Pixels/MKL, funneled	0.6822
Eigenfaces, original	0.6002

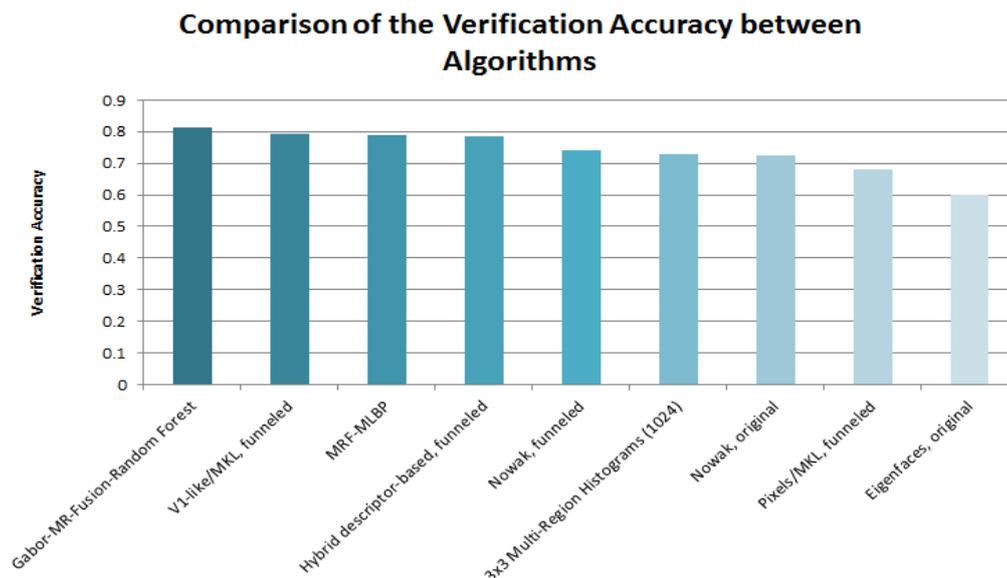


Figure 4.18: Comparison of Accuracy Rate among Algorithms on LFW Database

Based on Table 4.6 and Figure 4.18, it is found that the proposed algorithm outperforms the other algorithms in terms of recognition accuracy, standing at 81.28% accuracy using parameter settings in Table 4.4.

4.8.2.3 Receiver Operating Characteristics Curve

The use of ROC curve as biometric evaluation tool has several properties:

1. It demonstrates the relationship between true positive rate and false positive rate. The rise in true positive rate leads to the falling of the false positive rate
2. It is more ideal that the curve approaches y-axis and the top left corner of the ROC space.
3. If the curve approaches the straight diagonal line ($y = x$) of the ROC space, the performance of the algorithm is least accurate.
4. The area under the curve (AUC) is the indicator of accuracy level. The area of 1 implies that the performance is perfect. The area of 0.5 implies a very poor performance. The area denotes the ability of the algorithm to classify correctly.
5. It is the measure of true positive rate against the false positive rate. The reason of the inaccuracies of the classifier is that there is distribution overlap

(false positive and false negative distributions) between true negative distribution and true positive distribution. The ROC distribution curve is shown in Figure 4.19,.

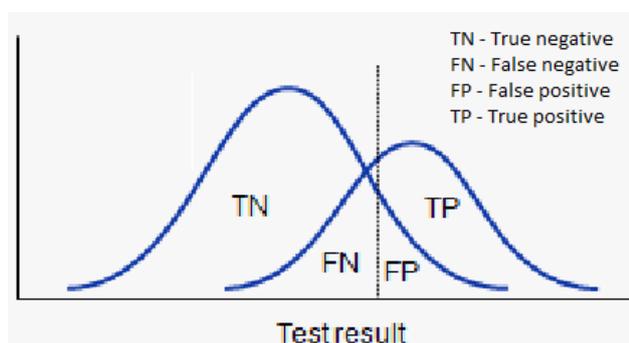


Figure 4.19: The ROC distribution curve

4.8.2.3.1 Comparison of ROC among existing algorithms

Table 4.7: Comparison of ROC AUC among algorithms on LFW database using the same parameter setting shown in Table 4.4 (Learned-Miller, 2016).

Algorithms	ROC (Highest at 1.0)
MRF-Fusion-CSKDA	0.9894
Gabor-MR-Fusion-Random Forest	0.9707
Spartans	0.9428
Pose Adaptive Filter (PAF)	0.9405
LBPNetLBP	0.9404
SA-BSIF, WPCA, aligned	0.9318
MRF-MLBP	0.8994
LHS, aligned	0.8107

LARK unsupervised, aligned	0.783
H-XS-40, 81x150, funneled	0.7547
GJD-BC-100, 122x225, funneled	0.7392
SD-MATCHES, 125x125, funneled	0.5407

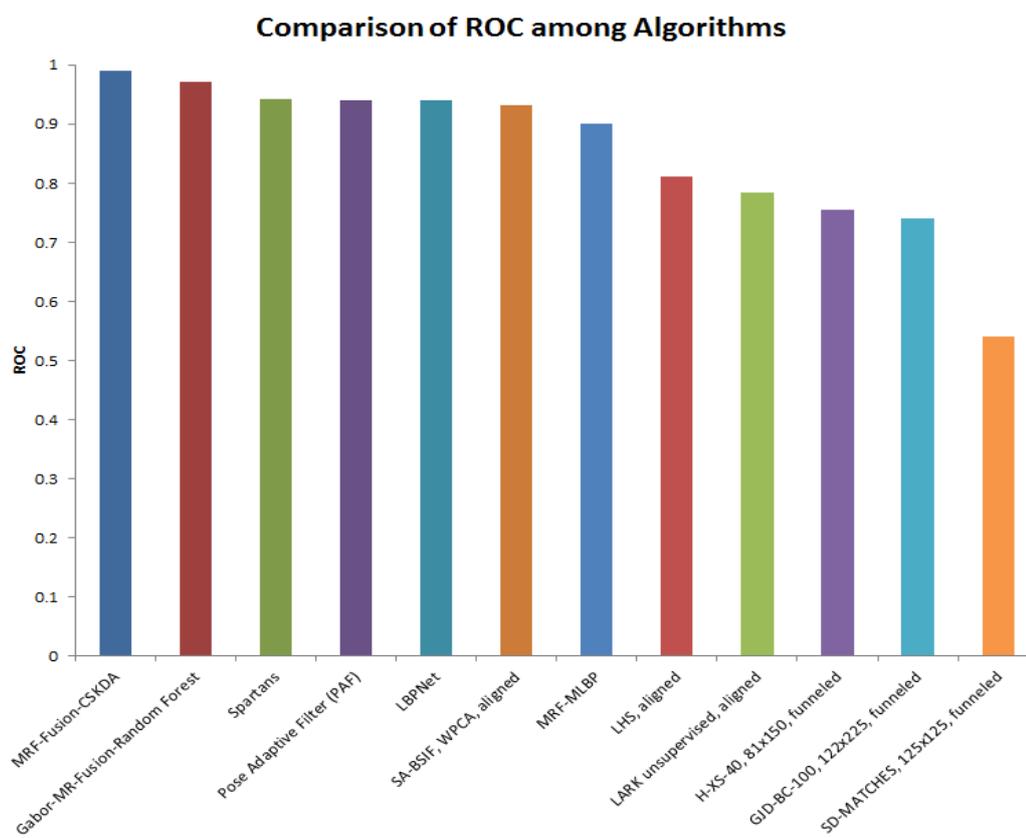


Figure 4.20: Comparison of ROC AUC among algorithms on LFW database

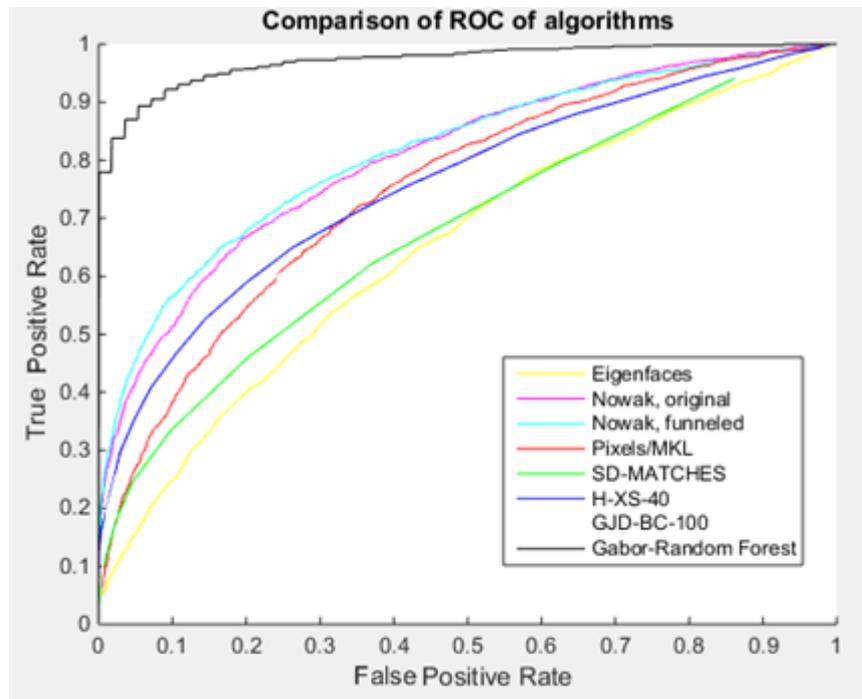


Figure 4.21: Visualization of ROC Curve of existing algorithms (only ROC of some algorithms are shown here)

Based on Figure 4.20, it is found that Gabor-MR-fusion with Random Forest performs second best among the existing algorithms in terms of ROC area under the curve (AUC) at 97.07%, outperforming every algorithm except the MRF-Fusion-CSKDA.

In Figure 4.21, it is shown that Gabor-MR-fusion with Random Forest ROC curve knee approaches closer to the top left corner of the ROC space than any other algorithms. The closer the ROC curve knee is to the top left corner of the ROC space, the higher accuracy of the algorithm gets.

4.8.2.4 Detection Error Tradeoff (DET) Curve

In using Detection Error Tradeoff (DET) as biometric evaluation tool, the classification involves the tradeoff between false negatives and false positives. Algorithm performs the best if the DET curve approaches the origin of the DET space. For LFW database, the area under the curve (AUC) of the DET curve stands

at 0.02456. This shows that the Gabor-MR-fusion with Random Forest algorithm gives good performance as the DET-AUC is low and is close to the ideal level of zero as shown in Figure 4.22.

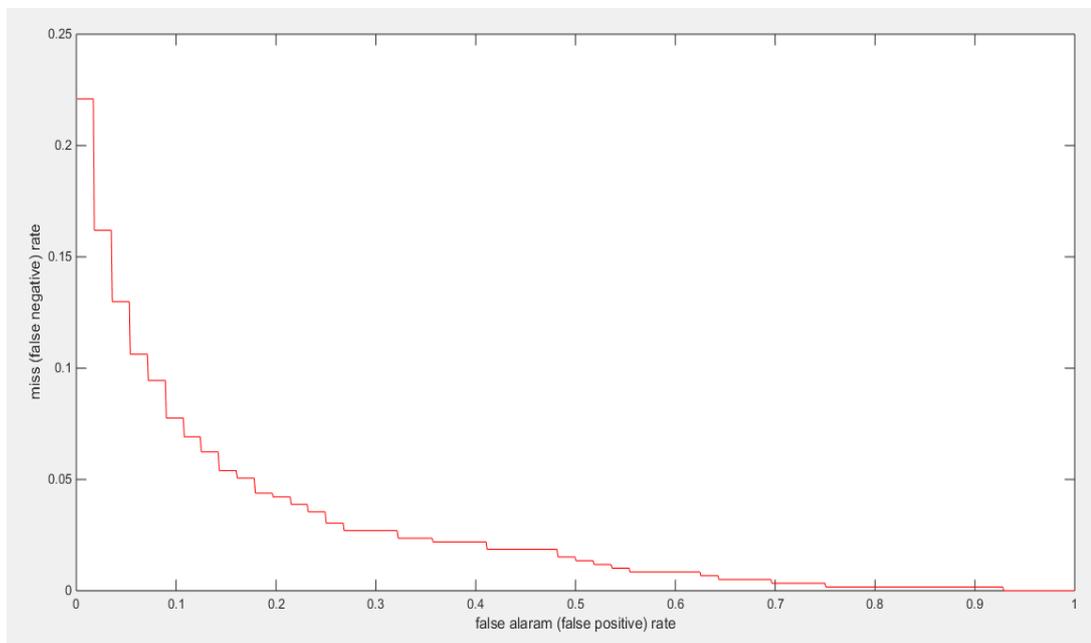


Figure 4.22: Detection Error Tradeoff Curve on LFW database

4.8.3 Unconstrained Facial Images (UFI)

4.8.3.1 Confusion Matrix

Figure 4.23 until Figure 4.25 show the confusion matrix of implemented method applied on the UFI database.

4.8.3.2 Recognition Rate or Classification Accuracy

Table 4.8: Comparison of Classification Accuracy among Algorithms on UFI Database (Ufi.kiv.zcu.cz., 2017)

Algorithm	Classification Accuracy (%)
Gabor-MR-Fusion-Random Forest	67.33
POEMHS	67.11
Enhanced Local Binary Patterns	65.28
FS-LBP	63.31
SIFT	61.32
LBPHS	55.04
MGM+LBP Edge-mapped	51.07
LDPHS	50.25

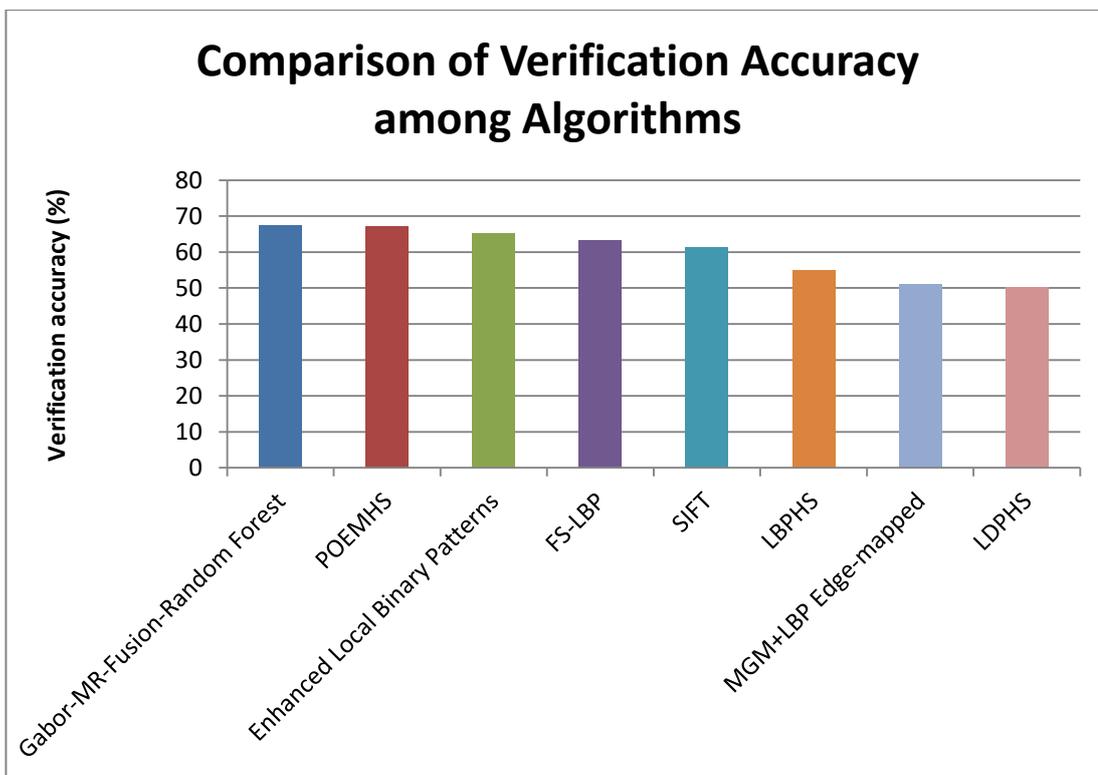


Figure 4.26: Comparison of Classification Accuracy among Algorithms on UFI Database

Based on the Table 4.8 and Figure 4.26, it is shown that Gabor-MR-Fusion-Random Forest obtains the highest classification accuracy at 67.33% in comparison with other reported existing algorithms.

4.8.3.3 Receiver Operating Characteristics (ROC) Curve

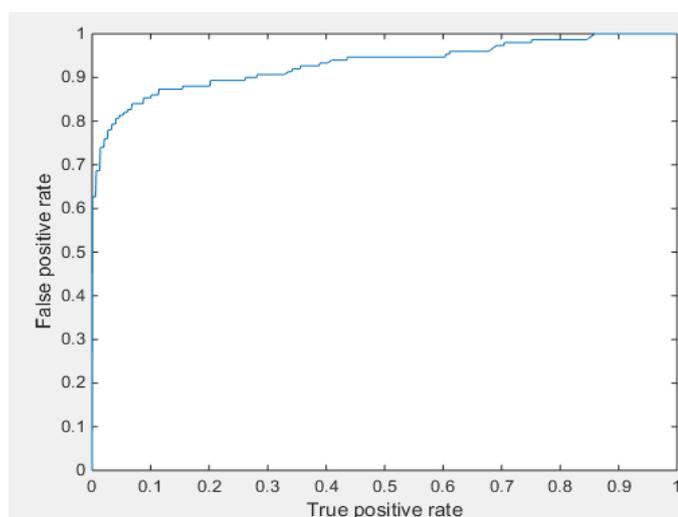


Figure 4.27: Visualization ROC Curve of Gabor-MR-fusion Random Forest on UFI Database

Table 4.9: Comparison of ROC AUC between Algorithms

Algorithm	ROC Area Under the Curve (%)
Gabor-MR-fusion with Random Forest algorithm	93.06
Hybrid of CBIR and SVM (Singh, et al., 2005)	95.42

In Table 4.9 and Figure 4.27, the implemented Gabor-MR-fusion with Random Forest algorithm gets 93.06% area under the curve (AUC) which is slightly lower than the Hybrid of CBIR and SVM. As assumed early, the higher the ROC AUC, the higher the accuracy of the algorithm.

4.8.3.4 Detection Error Tradeoff (DET) Curve

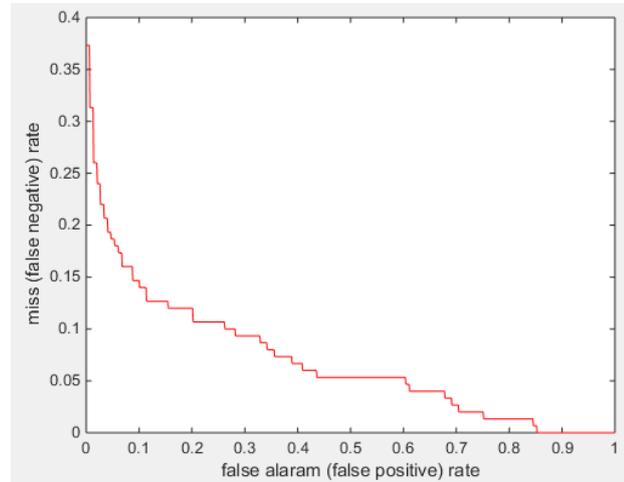


Figure 4.28: Visualization DET Curve of Gabor-MR-fusion Random Forest on UFI Database

According to Figure 4.28, Gabor-MR-fusion with Random Forest algorithm gets 6.99% area under the curve (AUC). The lower the DET AUC, the higher the accuracy of the algorithm.

4.8.4 Summary of the Comparison of Algorithms using LFW and UFI Databases

In summary, Gabor-MR-fusion with Random Forest obtains 81.28% and 67.33% of recognition rates and 97.07% and 93.06% of ROC-AUC on LFW and UFI databases. The implemented method performs better than several algorithms such as FS-LBP, LDPHS and SIFT etc. but performs poorer than Hybrid of CBIR and SVM and MRF-Fusion-CSKDA.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In conclusion, the implemented method in this project is Gabor-Maximum Response Filters with Random Forests Classifiers. The Gabor filters utilizes the features generated from magnitude and phase responses. Maximum Response filters exploits features generated by edge, bar and isotropic filters. The Random Forests is used as classifier. The biometric performance is evaluated on two unconstrained facial image databases which are Labelled Faces in the Wild (LFW) and Unconstrained Facial Images (UFI). The implemented method shows better performance than that of several existing algorithms such as Enhanced Local Binary Patterns and SIFT. This shows that the implemented face recognition method achieves good performance on facial images under uncontrolled environment which bears resemblance to the ambient conditions of the crime scenes.

5.2 Challenges and Recommendations for future work

There are many challenges in face recognition in criminal investigation. For example, the age factor and the presence of wrinkles can worsen the performance of face recognition. Furthermore, it is difficult to gather a dataset of images of the same individuals at different ages. The age invariant discriminative features and wrinkle feature masks could be explored in order to construct a model for age progression or synthesis (Ling, 2007).

Besides that, the occlusions of face due to the wearing of facial veil or sunglasses or the partial blocking of the surveillance camera by the tree branches can affect the face recognition performance. The part based technique is recommended for the case of partial occlusions where the face region is segmented into non-overlapping patches and the individual patches are evaluated (Azeem, 2014).

In future works, face recognition using deep learning neural networks can be further studied as deep learning has advantages in processing unstructured data such as visual images. It does not need the labelling of every face to discover facial patterns. Meanwhile, Random Forests used in this project requires the labelled data. Deep learning outperforms the Random Forests classifier in terms of prediction performance.

In deep learning, every pixel of the input and target images is vectorized. The deep learning model performs geometric transformation on the data and then does the mapping of input space onto target space. The transformation is more complex and requires higher computation costs than Gabor filter feature extraction (Blog.keras.io, 2017). Thus, the method of fusion between Gabor feature extraction and deep learning is suggested because this method trains the system to learn a small set of filters to produce an enhanced deep learning model (Luan, 2017). The fusion has the advantages of high accuracy of deep learning and the capability of Gabor filters in extracting the key features in a faster way. This could save computation costs and time.

REFERENCES

- Abdulrahman, M., Gwadabe, T.R., Abdu, F.J. and Eleyan, A., 2014, April. Gabor wavelet transform based facial expression recognition using PCA and LBP. In *Signal Processing and Communications Applications Conference (SIU), 2014 22nd* (pp. 2265-2268). IEEE.
- Azeem, A., Sharif, M., Raza, M. and Murtaza, M., 2014. A survey: Face recognition techniques under partial occlusion. *Int. Arab J. Inf. Technol.*, 11(1), pp.1-10.
- Bargavi, B.S. and Santhi, C., 2014. Global and local facial feature extraction using Gabor filters. *International Journal of Science, Engineering and Technology Research (IJSETR)*, 3(4), pp.1020-1023.
- Belhumeur, P.N., Hespanha, J.P. and Kriegman, D.J., 1997. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7), pp.711-720.
- Bhuiyan, A.-A., & Liu, C. H. (2007). On face recognition using gabor filters. *World academy of science, engineering and technology*, 28, 51-56.
- Bianconi, F. and Fernández, A., 2007. Evaluation of the effects of Gabor filter parameters on texture classification. *Pattern Recognition*, 40(12), pp.3325-3335.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.
- Brownlee, J. (2017). *An Introduction to Feature Selection - Machine Learning Mastery*. [online] *Machine Learning Mastery*. Available at: <https://machinelearningmastery.com/an-introduction-to-feature-selection/> [Accessed 26 Aug. 2017].
- Caenen, G. and Van Gool, L., 2004, June. Maximum response filters for texture analysis. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on* (pp. 58-58). IEEE.
- Cai, W., Li, Y. and Shao, X., 2008. A variable selection method based on uninformative variable elimination for multivariate calibration of near-infrared spectra. *Chemometrics and intelligent laboratory systems*, 90(2), pp.188-194.
- CNBC. (2017). Keen singer to police cell - Vietnamese suspect in Kim murder. [online] Available at: <https://www.cnn.com/2017/02/22/keen-singer-to-police-cell--vietnamese-suspect-in-kim-murder.html> [Accessed 26 Aug. 2017].
- Cortes, C. and Vapnik, V., 1995. Support-vector networks. *Machine learning*, 20(3), pp.273-297.
- Datta, A.K., Datta, M. and Banerjee, P.K., 2015. *Face Detection and Recognition: Theory and Practice*. CRC Press.
- Daugman, J.G., 1980. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision research*, 20(10), pp.847-856.

Daugman, J.G., 1988. Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7), pp.1169-1179.

Docs.opencv.org. (2017). Introduction to Support Vector Machines — OpenCV 2.4.13.3 documentation. [online] Available at: http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html [Accessed 26 Aug. 2017].

Geusebroek, J.M., Smeulders, A.W. and Van De Weijer, J., 2003. Fast anisotropic gauss filtering. *IEEE Transactions on Image Processing*, 12(8), pp.938-943.

Ghosal, V., Tikmani, P., & Gupta, P. (2009). Face classification using gabor wavelets and random forest. Paper presented at the Computer and Robot Vision, 2009. CRV'09. Canadian Conference on.

Han, Q.J., Wu, H.L., Cai, C.B., Xu, L. and Yu, R.Q., 2008. An ensemble of Monte Carlo uninformative variable elimination for wavelength selection. *Analytica chimica acta*, 612(2), pp.121-125.

Hassan, M. (2017). Iris recognition is Most Accurate Biometric Modality. [online] M2SYS Blog On Biometric Technology. Available at: <http://www.m2sys.com/blog/biometric-hardware/reliable-biometric-modality/> [Accessed 26 Aug. 2017].

Heyden, A. (2006). *Computer vision - ECCV 2002*. Berlin: Springer.

Ho, T. K. 1998. The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20, 832- 844.

Kohavi, R. and John, G.H., 1997. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2), pp.273-324.

Kohonen, T., 2012. *Self-organization and associative memory (Vol. 8)*. Springer Science & Business Media.

Kong, W. & Zhu, S. 2007. Multi-face detection based on downsampling and modified subtractive clustering for color images. *Journal of Zhejiang University-Science A*, 8, 72-78.

Kouzani, A.Z., Nahavandi, S. and Khoshmanesh, K., 2007, October. Face classification by a random forest. In *TENCON 2007-2007 IEEE Region 10 Conference* (pp. 1-4). IEEE.

Kovesi, P., 2000. Phase congruency: A low-level image invariant. *Psychological research*, 64(2), pp.136-148.

Learned-Miller, E., Huang, G.B., RoyChowdhury, A., Li, H. and Hua, G., 2016. Labeled faces in the wild: A survey. In *Advances in face detection and facial image analysis* (pp. 189-248). Springer International Publishing.

Lenc, L. and Král, P., 2015, October. Unconstrained Facial Images: Database for face recognition under real-world conditions. In *Mexican International Conference on Artificial Intelligence* (pp. 349-361). Springer International Publishing.

Lenc, L. and Král, P., 2016. Local binary pattern based face recognition with automatically detected fiducial points. *Integrated Computer-Aided Engineering*, 23(2), pp.129-139.

- Li, J., 2010. Classification/Decision Trees (I). Harrisburg: The Pennsylvania State University. Libpls.net. (2017). help_pls. [online] Available at: http://www.libpls.net/help/help_pls.html [Accessed 26 Aug. 2017].
- Ling, H., Soatto, S., Ramanathan, N. and Jacobs, D.W., 2007, October. A study of face recognition as people age. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on (pp. 1-8). IEEE.
- Liu, C., & Wechsler, H. (2001). A Gabor feature classifier for face recognition. Paper presented at the Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on.
- Liu, W. and Wang, Z., 2006, August. Facial expression recognition based on fusion of multiple Gabor features. In Pattern Recognition, 2006. ICPR 2006. 18th International Conference on (Vol. 3, pp. 536-539). IEEE.
- Lorena, A.C. and De Carvalho, A.C., 2008. Evolutionary tuning of SVM parameter values in multiclass problems. *Neurocomputing*, 71(16), pp.3326-3334.
- Lorena, A.C., Jacintho, L.F., Siqueira, M.F., De Giovanni, R., Lohmann, L.G., De Carvalho, A.C. and Yamamoto, M., 2011. Comparing machine learning classifiers in potential distribution modelling. *Expert Systems with Applications*, 38(5), pp.5268-5275.
- Luan, S., Zhang, B., Chen, C., Cao, X., Ye, Q., Han, J. and Liu, J., 2017. Gabor Convolutional Networks. arXiv preprint arXiv:1705.01450.
- Mao, Z., Cai, W. and Shao, X., 2013. Selecting significant genes by randomization test for cancer classification using gene expression data. *Journal of biomedical informatics*, 46(4), pp.594-601.
- Mehmood, T., Liland, K.H., Snipen, L. and Sæbø, S., 2012. A review of variable selection methods in partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 118, pp.62-69.
- Ojala, T., Pietikainen, M. and Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7), pp.971-987.
- Peterkovesi.com. (2017). Log-Gabor Filters. [online] Available at: <http://www.peterkovesi.com/matlabfns/PhaseCongruency/Docs/convexpl.html> [Accessed 26 Aug. 2017].
- Reunanen, J., 2003. Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, 3(Mar), pp.1371-1382.
- Rodriguez-Galiano, V., Ghimire, B., Rogan, J., Chica-Olmo, M. and Rigol-Sanchez, J. (2012). An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 67, pp.93-104.
- Samaria, F., & Young, S. (1994). HMM-based architecture for face identification. *Image and vision computing*, 12(8), 537-543.

- Seo, H.J. and Milanfar, P., 2011. Face verification using the lark representation. *IEEE Transactions on Information Forensics and Security*, 6(4), pp.1275-1286.
- Sharif, M., Khalid, A., Raza, M. and Mohsin, S., 2011. Face Recognition using Gabor Filters. *Journal of Applied Computer Science & Mathematics*, (11).
- Singh, A., Sohoni, P. and Kumar, M., 2015. A Hybrid Approach for CBIR using SVM Classifier, Partical Swarm Optimizer with Mahalanobis Formula. *International Journal of Computer Applications*, 111(12).
- Srivastava, T. (2017). Introduction to KNN, K-Nearest Neighbors : Simplified. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2014/10/introduction-k-neighbours-algorithm-clustering/> [Accessed 20 Aug. 2017].
- Štruc, V. and Pavešić, N., 2009. Gabor-based kernel partial-least-squares discrimination features for face recognition. *Informatika*, 20(1), pp.115-138.
- Struc, V. and Pavesic, N., 2010. From Gabor Magnitude to Gabor Phase Features: Tackling the Problem of Face Recognition under Severe Illumination Changes. In *Face Recognition. InTech*.
- Štruc, V., & Pavešić, N., 2009. Principal gabor filters for face recognition. Paper presented at the *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*.
- Sun, W., Sun, N., Guo, B., Jia, W. and Sun, M., 2016. An auxiliary gaze point estimation method based on facial normal. *Pattern Analysis and Applications*, 19(3), pp.611-620.
- Tan, X. and Triggs, B., 2007. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *Analysis and modeling of faces and gestures*, pp.168-182.
- Turk, M. A., & Pentland, A. P. (1991). Face recognition using eigenfaces. Paper presented at the *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*.
- Turner, M.R., 1986. Texture discrimination by Gabor functions. *Biological cybernetics*, 55(2), pp.71-82.
- Ufi.kiv.zcu.cz. (2017). Unconstrained Facial Images. [online] Available at: <http://ufi.kiv.zcu.cz/> [Accessed 26 Aug. 2017].
- Viola, P. and Jones, M.J., 2004. Robust real-time face detection. *International journal of computer vision*, 57(2), pp.137-154.
- Vis-cs.umass.edu. (2017). LFW Face Database : Main. [online] Available at: <http://vis-www.cs.umass.edu/lfw/> [Accessed 26 Aug. 2017].
- Vitomir, Š. and Nikola, P., 2010. The complete gabor-fisher classifier for robust face recognition. *EURASIP Journal on Advances in Signal Processing*, 2010(1), p.847680.
- Wang, Z., 2015. Variable selection for industrial process modeling and monitoring (Doctoral dissertation).

- Xia, W., Yin, S. and Ouyang, P., 2013. A high precision feature based on LBP and Gabor theory for face recognition. *Sensors*, 13(4), pp.4499-4513.
- Xiao, P., Fenga, X., Zhaoa, S., Xieb, S., Wanga, P. and Badawia, R., 2007, June. Applying texture marker-controlled watershed transform to the segmentation of IKONOS image. In *Proc. of SPIE Vol* (Vol. 6752, pp. 67520W-1).
- Zhang, B., Shan, S., Chen, X. and Gao, W., 2007. Histogram of gabor phase patterns (hgpp): A novel object representation approach for face recognition. *IEEE Transactions on Image Processing*, 16(1), pp.57-68.
- Zhao, W., Chellappa, R. and Krishnaswamy, A., 1998, April. Discriminant analysis of principal components for face recognition. In *Automatic Face and Gesture Recognition*, 1998. Proceedings. Third IEEE International Conference on (pp. 336-341). IEEE.
- Zhou, S.R., Yin, J.P. and Zhang, J.M., 2011. Local binary pattern (LBP) and local phase quantization (LBQ) based on Gabor filter for face representation. *Neurocomputing*, 116, pp.260-264.
- Zhu, J., Zou, H., Rosset, S. and Hastie, T., 2009. Multi-class adaboost. *Statistics and its Interface*, 2(3), pp.349-360.

APPENDIX

APPENDIX A: COMPUTER SPECIFICATION

System	<hr/>	
Processor:	Intel(R) Core(TM) i7-3537U CPU @ 2.00GHz 2.50 GHz	
Installed memory (RAM):	6.00 GB (5.87 GB usable)	
System type:	64-bit Operating System, x64-based processor	
Pen and Touch:	No Pen or Touch Input is available for this Display	

APPENDIX B: MATLAB CODES

main_fused.m

```

clear all;

prompt = {'MODE';'test_mode';'DATABASE';'TRAIN METHOD'};
name = 'Input';
formats = struct('type', {}, 'style', {}, 'items', {}, ...
    'format', {}, 'limits', {}, 'size', {});

formats(1,1).type = 'list';    % mode
formats(1,1).style = 'popupmenu';
formats(1,1).items = {'TRAIN','TEST'};
formats(2,1).type = 'list';
formats(2,1).style = 'popupmenu';
formats(2,1).items = {'Manual','Test All'};
formats(3,1).type = 'list';
formats(3,1).style = 'popupmenu';
formats(3,1).items = {'czech','lfw'};
formats(4,1).type = 'list';
formats(4,1).style = 'popupmenu';
formats(4,1).items = {'GABOR', 'PHASE','MR'};
defaultanswer = {1, 1, 1, 1};

[answer, Canceled] = inputsdlg(prompt, name, formats, defaultanswer);
if (Canceled)
    dbquit
end

MODE = {'TRAIN','TEST'};
test_mode = {'Manual','Test All'};
DATABASE = {'czech','lfw'};
METHOD = {'GABOR', 'PHASE','MR'};

```

```

mode = MODE{answer{1}};
test_mode = test_mode{answer{2}};

switch mode
case 'TRAIN'
    disp('Training Mode');
    method_name = METHOD{answer{4}};
    fprintf(' method_name: %s \n', method_name);
    database_name = DATABASE{answer{3}};
    fprintf(' database_name: %s \n', database_name);

    if strcmp(database_name, 'lfw')
        img_size = [100 100];
    else
        img_size = [100, 100]; %[128 128];
    end

    selecttype = 'all';
    enablesave = 1;
    opt.method = struct('name',method_name,'psize', img_size);
    opt.param.gabor = struct('Downsampling', 64); %[100]); % 200 X 200 X 12 X 8 / 384 =
10000
    opt.RF = struct('NTrees',500,'parallel',1,'RandStream',0,'SelectBest',0);
    opt.database = struct('name', database_name,'fileselect',selecttype);
    opt.database = database_initialization(opt.database);
    rf = TrainRF_fused(opt);
    if (strcmp(method_name,'GABOR'))
        save('TrainRF_output_Gabor_60.mat', 'rf') ;
    elseif (strcmp(method_name,'PHASE'))
        save('TrainRF_output_Phase_60.mat', 'rf') ;
    elseif (strcmp(method_name,'MR'))
        save('TrainRF_output_MR_60.mat', 'rf') ;
    else

```

```

        % do nothing
    end

case 'TEST'
    database_name = DATABASE{answer{3}};
    disp('Loading Mat file');
    fprintf('test_mode: %s \n', test_mode);

    if (strcmp(database_name, 'czech'))
        mat_file_gabor = 'TrainRF_output_Gabor_26.mat';
    elseif (strcmp(database_name, 'lfw'))
        mat_file_gabor = 'TrainRF_output_Gabor_23.mat';
    end

    rf_gabor = loadRF(mat_file_gabor);
    disp('Testing Mode');

    if (strcmp(test_mode, 'Manual' ))
        tic
        if (strcmp(database_name, 'czech'))
            temp = load('TrainRF_opt_Treebagger_Output_Gabor_26.mat');
            b = temp.b;
            temp = load('TrainRF_full_Treebagger_Output_Phase_26.mat');
            b1 = temp.bb;
            temp = load('TrainRF_opt_Treebagger_Output_MR8_48.mat');
            b2 = temp.b;
            temp = load('TrainRF_SortedVariable_UVE_Gabor_26.mat');
            Gabor_SortedVariable = temp.UVE.SortedVariable;
            temp = load('TrainRF_SortedVariable_UVE_MR8_48.mat');
            MR8_SortedVariable = temp.UVE.SortedVariable;
        elseif (strcmp(database_name, 'lfw'))
            temp = load('TrainRF_opt_Treebagger_Output_Gabor_23.mat');
            b = temp.b;

```

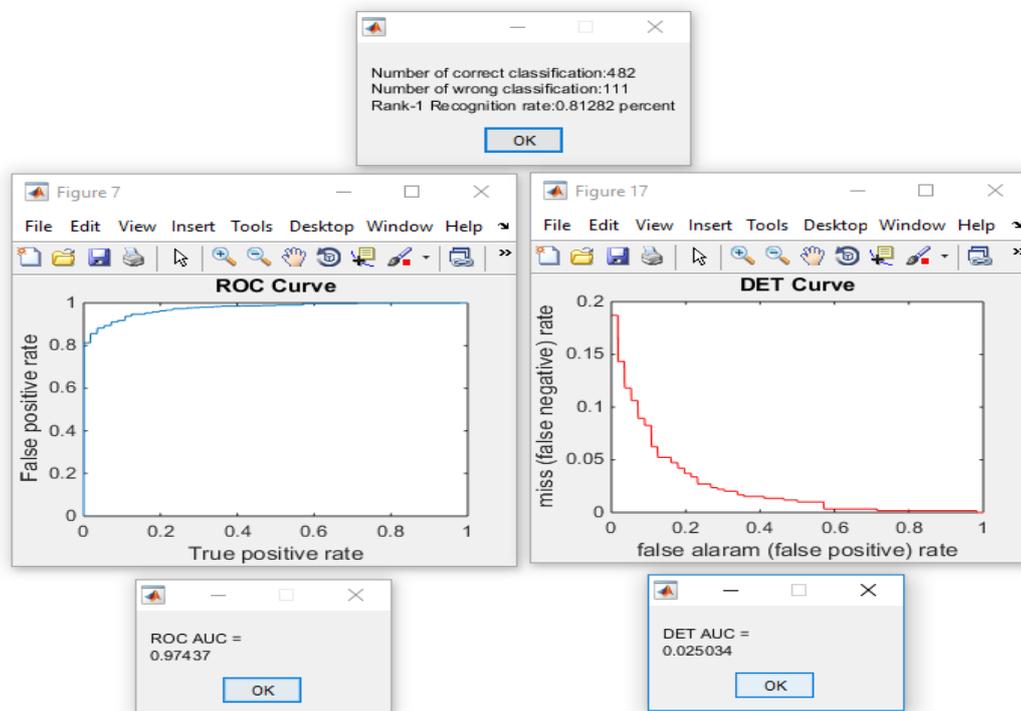
```

temp = load('TrainRF_full_Treebagger_Output_Phase_23.mat');
b1 = temp.bb;
temp = load('TrainRF_opt_Treebagger_Output_MR8_36.mat');
b2 = temp.b;
temp = load('TrainRF_SortedVariable_UVE_Gabor_23.mat');
Gabor_SortedVariable = temp.UVE.SortedVariable;
temp = load('TrainRF_SortedVariable_UVE_MR8_36.mat');
MR8_SortedVariable = temp.UVE.SortedVariable;
end
toc

fprintf('Finished loading \n');
flag = 1;
while (flag)
    result = testRF_fused(rf_gabor, test_mode, database_name, b, b1, b2,
Gabor_SortedVariable, MR8_SortedVariable );
    choice = questdlg('Would you like to continue testing?', 'Testing question',
'Yes', 'No', 'Yes');
    switch choice
        case 'Yes'
            flag = 1;
        case 'No'
            flag = 0;
    end
end
elseif (strcmp(test_mode,'Test All' ))
    result = testRF_fused_2(rf_gabor, test_mode, database_name);
end
save('test_result_Complete_70.mat', 'result');
otherwise
end
disp('DONE');

```

APPENDIX C: USER INTERFACE

LFWUFI