# Real Fingerprint Detection System (RFDS) Based on Image Quality Measures and Six Classifiers

Abdurahman Ibrahim Mriheel

Submitted to the Institute of Graduate Studies and Research in partial fulfillment of the requirements for the degree of

> Master of Science in Computer Engineering

Eastern Mediterranean University February 2020 Gazimağusa, North Cyprus Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ali Hakan Ulusoy Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Işık Aybay Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Alexander Chefranov Supervisor

**Examining Committee** 

1. Assoc. Prof. Dr. Alexander Chefranov

2. Assoc. Prof. Dr. Önsen Toygar

3. Asst. Prof. Dr. Mehtap Köse Ulukök

## ABSTRACT

Fingerprint detection in biometrics is an important field of study in our new modern world, many forensic departments around the world use fingerprints as the key to detect criminals and bring them justice. To improve the accuracy of fingerprint detection system we implemented a Real Fingerprint Detection System (RFDS) that has high performance level of detecting real and fake fingerprint images. In this thesis, we present an RFDS system based on image quality measures (IQM's) to detect real fingerprint images and fake fingerprint images. We performed different RFDS experiments with 25, 10, 15, and 5 IQM's; they showed sufficient quality of real and fake fingerprint images detection. We compared our RFDS using 25, 15, 10, and 5 IQM's with RFDS that has used 25 IQMs. Based on the comparison in this thesis we can conclude that the best result from all these RFDS is the one with 25 IQMs, because it's HTER score is the minimum one with 0.3%, and the worst RFDS is the one with the 15 IQMs which has the maximum HTER score with 14.8%.

**Keywords**: Biometrics, Image Quality Measure, Real and Fake Fingerprint Image, Classifier.

Biyometride parmak izi tespiti, yeni modern dünyamızda önemli bir çalışma alanıdır, dünyadaki birçok adli departman, suçluları tespit etmek ve adalet getirmek için parmak izi kullanmaktadır. Parmak izi algılama sisteminin doğruluğunu artırmak için, gerçek ve sahte parmak izi görüntülerini algılamada yüksek performans seviyesine sahip bir Gerçek Parmak İzi Algılama sistemi (RFDS) uyguladık. Bu tezde, gerçek parmak izi görüntülerini ve sahte parmak izi görüntülerini tespit etmek için görüntü kalitesi ölçümlerine (IQM'ler) dayanan bir RFDS sistemi sunuyoruz. 25, 15, 10 ve 5 IQM'lerle farklı RFDS deneyleri gerçekleştirdik; yeterli kalitede gerçek ve sahte parmak izi görüntüleri algılama özelliği gösterdiler. 25, 15, 10 ve 5 IQM kullanarak RFDS'imizi 25 IQM kullanan RFDS ile karşılaştırdık. Bu tezdeki karşılaştırmaya dayanarak, tüm bu RFDS'lerden en iyi sonucun 25 IQM'ye sahip olduğu sonucuna varabiliriz, çünkü HTER puanı % 0.3 ile en düşük olanıdır ve en kötü RFDS ise %14.8 HTER puanı ile 15 IQM'ye sahip olanıdır.

Anahtar Kelimeler: Biyometri, Görüntü Kalitesi Ölçüsü, Gerçek Ve Sahte Parmak Izi Görüntüsü, Sınıflandırıcı.

# **DEDECATION**

To My Family

## ACKNOWLEDGMENT

I would like to thank my supervisor (Assoc. Prof. Dr. Alexander Chefranov) who gave me the golden opportunity to do this wonderful project on the topic (Real Fingerprint Detection System (RFDS) Based on Image Quality Measures and Six Classifiers), which also helped me in doing a lot of research and I came to know about so many new things, I am thankful to him.

Secondly, I would like also to express my special thanks of gratitude to my parents and friends who helped me in finalizing this project within the limited time frame

# TABLE OF CONTENTS

ABSTRACTiii
ÖZiv
DEDECATION
ACKNOWLEDGMENTvi
LIST OF TABLESxi
LIST OF FIGURESxii
LIST OF ABBREVIATIONS
1 INTRODUCTION
1.1 Definition of Common Biometric Terms [3]
1.2 Fingerprint Detection Systems
1.3 Key Terms Used in Fingerprint Images Analysis
1.4 Summary
2 ANALYSIS F KNOWN RFDS AND PROBLEM DEFINITION
2.1 Analysis of Known RFDS5
2.1.1 Image Quality Assessment for Fake Biometric Detection: Application to
Iris, Fingerprint, And Face Recognition [6]5
2.1.2 Full Reference (FR) Image Quality Measures Used In [6]6
2.1.3 Classifiers Used in [6] and Our RFDS12
2.1.3.1 RFDS Experiments In [6]15
2.1.4 Fingerprint Liveness Detection Based on Quality Measures [27]17
2.1.4.1 RFDS Experiments In [27]17
2.1.5 A High-Performance Fingerprint Liveness Detection Method Based on
Quality Related Features [28]18

2.1.5.1 RFDS Experiments In [28]19
2.1.6 Fingerprint Identification in Biometric Security Systems. International
Journal of Computer and Electrical Engineering [29]20
2.1.6.1 RFDS Experiments In [29]21
2.1.7 Fingerprint Spoof Detection Using Quality Features [30]21
2.1.7.1 RFDS Experiments In [30]21
2.2 Problem Definition
2.3 Summary
3 DESIGN AND IMPLEMENTATION OF RFDS24
3.1 Design of RFDS
3.2 Implementation of RFDS
3.2.1 Implementation and Testing of Gaussian Filter25
3.2.2 Implementation and Testing of Feature Extraction
3.2.2.1 Implementation and Testing of MSE (2.1)
3.2.2.2 Implementation and Testing of PSNR (2.2)
3.2.2.3 Implementation and Testing of SNR (2.3)
3.2.2.4 Implementation and Testing of SC (2.4)
3.2.2.5 Implementation and Testing of MD (2.5)
3.2.2.6 Implementation and Testing of AD (2.6)
3.2.2.7 Implementation and Testing of NAE (2.7)
3.2.2.8 Implementation and Testing of RAMD (2.8)
3.2.2.9 Implementation and Testing of NCC (2.9)
3.2.2.10 Implementation and Testing of TED (2.10)
3.2.2.11 Implementation and Testing of TCD (2.11)
3.2.2.12 Implementation and Testing of Spectral Phase Error (SPE)41

3.2.2.13 Implementation and Testing of SME (2.13)
3.2.2.14 Implementation and Testing of Gradient Phase Error (GPE) 47
3.2.2.15 Implementation and Testing of Gradient Magnitude Error (GME)
3.2.2.16 Implementation and Testing of Mean Angle Similarity (MAS). 52
3.2.2.17 Implementation and Testing of Mean Angle Similarity (MAMS)
3.2.2.18 Implementation and testing of Laplacian Mean Squared Error
(LMSE)
3.2.3 Implementation of Training Structure
3.2.4 Implementation of The Classification Process
3.3 Development of A Method of Defining the Best IQMs67
3.3.1 Implementation and Testing of The Method of Defining the Best IQMs 69
3.3.1.1 Calculation of Defining the Best 5 IQMs70
3.4 Summary
4 EXPERIMENTS ON RFDS
4.1 Experimental Setup75
4.2 Database Used
4.3 Classification Methods Used76
4.4 Code Parts for Experiments Conducting76
4.5 Experimental Results
4.5.1 Experiments on the RFDS Training and Classification with 15 IQMs 77
4.5.2 Experiments on The Rfds Training and Classification with Best 10 IQMs

# LIST OF TABLES

Table 1: RFDS [6] results using three different type of fingerprint sensors
Table 2: The results for each sensor [27].    18
Table 3: Performance result of the two databases used in [28]
Table 4: Training results with 15 IQMs77
Table 5: Classification results with 15 IQMs
Table 6: Training results with best 10 IQMs.    79
Table 7: Classification results with best 10 IQMs.    79
Table 8: Training results with best 5 IQMs.    80
Table 9: Classification results with best 5 IQMs.    81
Table 10: Training results with 25 IQMs.    81
Table 11: Classification results with 25 IQM's.    82
Table 12: Result of comparison HTER (%) between our RFDS and RFDS [6] 83

# **LIST OF FIGURES**

Figure 1: Fingerprints ridge ending (a) and ridge bifurcation (b) [5]4
Figure 2: The boundary between two classes (Hyper $Plane(\hat{W})$ )
Figure 3: 'S' shaped logistic function
Figure 4: A new cell (fingerprint image) with unknown category to classify
Figure 5: Average classification error (ACE) results in [30]22
Figure 6: RFDS training and detection structure
Figure 7: Gaussian filter code and the Gaussian filter obtained from it
Figure 8: Result of the original image with Gaussian filter applied to it
Figure 9: MATLAB result of MSE calculation
Figure 10: MATLAB result of PSNR calculation
Figure 11: MATLAB result of SNR calculation
Figure 12: MATLAB result of SC calculation
Figure 13: MATLAB result of MD calculation
Figure 14: MATLAB result of AD calculation
Figure 15: MATLAB result of NAE calculation
Figure 16: MATLAB result of RAMD calculation with R=7
Figure 17: MATLAB result of NCC calculation
Figure 18: Matlab result for calculating TED of fingerprint image
Figure 19: MATLAB result of TCD calculation
Figure 20: MATLAB result of SPE calculation
Figure 21: MATLAB result of SME calculation
Figure 22: MATLAB result of GPE calculation
Figure 23: MATLAB result of GME calculation

Figure 24: MATLAB result of MAS calculation
Figure 25: MATLAB result of MAMS calculation
Figure 26: Matlab result of LMSE calculation
Figure 27: Table which has all the fingerprint image results of IQMS we used 59
Figure 28: The arrow shows a button used to call the classification learner app in
MATLAB
Figure 29: Start new session to import results
Figure 30: Selection of our dataset, predictors and response in MATLAB61
Figure 31: Selection of the classifier for our table imported and training result 62
Figure 32: Exporting training model of 100 real and fake fingerprint images
Figure 33: After exporting the training model, we can use it to make prediction by the
code provided in the command window highlighted with a rectangle
Figure 34: The highlighted rectangle shows the code used to classify images according
to our exported training model
Figure 35: Results of 10 real and fake images classified correctly using the
classification code
Figure 36: Fingerprint images used in our expirments, (A) fake fingerprint image from
the file" U01_F_Fc_Li_01.Bmp", (B) real fingerprint image form the file"
U01_O_Fc_Li_01.Bmp"67
Figure 37: Numerical axis for MSE (A) And MD (B), showing MSE of fake (real)
fingerprint images. And the respective averages calculated by (3.16) and (3.17)67
Figure 38: The list of the best IQMs (in the red rectangle)73
Figure 39: List of our best 15,10,5 IQMs used in our RFDS expirments74
Figure 40: QDA training result of 100 fingerprint images using 15 IQMs 120
Figure 41: Linear SVM training result of 100 fingerprint images using 15 IQMs. 120

Figure 42: Quadratic SVM training result of 100 fingerprint images using 15 IQMs.
Figure 43: Fine KNN training result of 100 fingerprint images using 15 IQMs 121
Figure 44: Logistic Regression training result of 100 fingerprint images using 15
IQMs
Figure 45: LDA classification result of 1000 fingerprint images using 15 IQMs 123
Figure 46: QDA classification result of 1000 fingerprint images using 15 IQMs123
Figure 47: Linear SVM classification result of 1000 fingerprint images using 15 IQMs.
Figure 48: Quadratic SVM classification result of 1000 fingerprint images using 15
IQMs
Figure 49: Fine KNN classification result of 1000 fingerprint images using 15 IQMs.
Figure 50: Logistic Regression classification of 1000 fingerprint images using 15
IQMs
Figure 51: LDA training result of 100 fingerprint images using 10 IQMs126
Figure 52: QDA training result of 100 fingerprint images using 10 IQMs 126
Figure 53: Linear SVM training result of 100 fingerprint images using 10 IQMs 127
Figure 54: Quadratic SVM training result of 100 fingerprint images using 10 IQMs.
Figure 55: Fine KNN training result of 100 fingerprint images using 10 IQMs 128
Figure 56: Logistic Regression training result of 100 fingerprint images using 10
IQMs
Figure 57: LDA Classification result of 1000 fingerprint images using 10 IQMs 129

Figure 58: QDA Classification Result of 1000 Fingerprint Images Using 10 IQMs.
Figure 59: Linear SVM classification result of 1000 fingerprint images using 10 IQMs.
Figure 60: Quadratic SVM classification result of 1000 fingerprint images using 10
IQMs
Figure 61: Fine KNN classification result of 1000 fingerprint images using 10 IQMs.
Figure 62: Logistic Regression classification result of 1000 fingerprint images using
10 IQMs
Figure 63: LDA training result of 100 fingerprint images using 5 IQMs132
Figure 64: QDA training result of 100 fingerprint images using 5 IQMs
Figure 65: Linear SVM training result of 100 fingerprint images using 5 IQMs133
Figure 66: Quadratic SVM training result of 100 fingerprint images using 5 IQMs.
Figure 67: Fine KNN training result of 100 fingerprint images using 5 IQMs 134
Figure 68: Logistic Regression training result displayed for 100 fingerprint images
using 5 IQMs134
Figure 69: LDA classification result of 1000 fingerprint images using 5 IQMs 135
Figure 70: QDA Classification of 1000 Fingerprint Images Using 5 IQMs135
Figure 71: Linear SVM LDA classification result of 1000 fingerprint images using 5
IQMs
Figure 72: Quadaratic SVM classification result of 1000 fingerprint images using 5
IQMs

Figure 73: Fine KNN classification result of 1000 fingerprint images using 5 IQMs.
Figure 74: Logistic Regression classification result of 1000 fingerprint images using 5
IQMs
Figure 75: LDA training result of 100 fingerprint images using 25 IQMs
Figure 76: QDA Training Result of 100 Fingerprint Images Using 25 IQMs 138
Figure 77: Linear SVM training result of 100 fingerprint images using 25 IQMs. 139
Figure 78: Quadaratic SVM training of 100 fingerprint images using 25 IQMs 139
Figure 79: Fine KNN training result of 100 fingerprint images using 25 IQMs 140
Figure 80: Logistic Regression training result of 100 fingerprint images using 25
IQMs
Figure 81: LDA classification result of 1000 fingerprint images using 25 IQMs 141
Figure 82: QDA classification result of 1000 fingerprint images using 25 IQMs141
Figure 83: Linear SVM classification result of 1000 fingerprint images using 25 IQMs.
Figure 84: Quadaratic SVM classification result of 1000 fingerprint images using 25
IQMs
Figure 85: Fine KNN classification result of 1000 fingerprint images using 25 IQMs.
Figure 86: Logistic Regression classification result of 1000 fingerprint images using
25 IQMs

# LIST OF ABBREVIATIONS

ACE	Average Classification Error
AD	Average Difference
FAR	False Accept Rate
FFR	False Fake Rate
FFR	False Fake Rate
FLR	False Living Rate
FR	Full Reference
FR-IQA	Full Reference Image Quality Assessment
GME	Gradient Magnitude Error
GPE	Gradient Phase Error
HTER	Half Total Error Rate
IQM	Image Quality Measures
LDA	Linear Discriminant Analysis
LMSE	Laplacian Mse
MAS	Mean Angle Similarity
MD	Maximum Difference
MSE	Mean Squared Error
NAE	Normalized Absolute Error
NCC	Normalized Cross Correlation
NR	No Reference
NR-IQA	No Reference Image Quality Assessment
PSNR	Peak Signal To Noise Ratio

QDA	Quadratic Discriminant Analysis
RAMD	R-Averaged Md
RFDS	Real Fingerprint Detection System
SC	Structural Content
SME	Spectral Magnitude Error
SNR	Signal To Noise Ratio
SPE	Spectral Phase Error
TCD	Total Corner Difference
TED	Total Edge Difference

## **Chapter 1**

## INTRODUCTION

Biometric recognition system is a mixture of hardware and software means used to differentiate and identify individuals [1]. Biometric recognition system obtains data such as fingerprint image of an individual, and then applies comparison to the image against the stored fingerprint images in the database. If the input fingerprint image matches one of the available images in the database, the system will accept it as **Real** image, otherwise it will be considered as **Fake** and will be rejected by the system. A biometric recognition system offers a natural and consistent solution to the problems of individual identification [1].

We can categorize the biometric methods used by biometric systems into two classes [2]:

1. Physiological-based methods use face, fingerprint, voice, retinal, DNA analysis, and measure the physiological characteristics of a person.

2. Behavior-based methods use signature, key stroke analysis and measure behavioral characteristics.

Biometric recognition systems can work in two modes:

• Identification Mode: The system recognizes a person searching a large database of registered ones for a match.

• Verification Mode: The system confirms a person's claimed identity from his previous registered pattern.

## 1.1 Definition of Common Biometric Terms [3]

• Enrollment or Registration: The individual's biometric data are obtained, processed and saved as a template for ongoing usage in a biometric system; this is known as enrollment, or registration process. The template will be used for identification or verification in further stages.

• **Biometric Data:** Any data related to the physiological and behavioral characteristics of an individual is known as biometric data, which can also be referred as raw biometric data.

• **Presentation:** In this procedure, the individual presents his or her biometric data to the hardware that is used to gather data. For example, placing your finger on a biometrical sensor.

• Feature Extraction: The procedure of finding and encoding unique characteristics from biometric data in order to produce a template is known as feature extraction. Feature extraction takes place during enrollment and verification stage.

• Image Quality Measures (IQM's): Mathematical measures applied to identify the quality of the image such as its color accuracy, noise, and sharpness, the results we get from the iqms assist us to identify or verify if the image is real or fake [4].

• Classification: A saved template will be compared with other templates during identification or verification stage. Based on the comparison we can classify if the individual is verified or not.

## **1.2 Fingerprint Detection Systems**

Fingerprint is unique for every individual; they are mainly used for identifying individuals. Real fingerprint detection systems (RFDS) are one of the most known biometric systems used in many applications such as border control and forensic department. RFDS are systems that identify or verify any fingerprint image fed to the system as real or fake image. Rfds systems are described in [6], [18], [19], [20], and [21]. RFDS performance is very critical, any failure can cause harm to individuals and governments. For example, detecting the innocent individual's fingerprint as the criminal, or passing the border control of a country with another individual's fingerprint, and system will detect it as true and make you pass the border control. These kinds of problems shouldn't occur as they will cause injustice and security problems to individuals and the governments. We will show different RFDS algorithms and implement the best of them, in which we will try to reduce the amount of wrong fingerprint detection as minimum as possible.

### **1.3 Key Terms Used in Fingerprint Images Analysis**

1. Ridge ending: the point, at which the ridge terminates, shown in figure 1, a.

Ridge bifurcation: when one ridge splits in-to two or more ridges, shown in figure
 b.

3. Minutiae: major features of a fingerprint like ridge endings and ridge bifurcations.

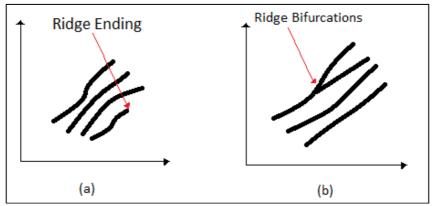


Figure 1: Fingerprints ridge ending (a) and ridge bifurcation (b) [5].

## **1.4 Summary**

In this thesis, we implement a real fingerprint detection system (RFDS) using feature extraction and classification algorithm to classify the registered or enrolled fingerprint image. In Chapter 2, we review known fingerprint detection systems and define the problems. In Chapter 3, we show the general diagram of the RFDS, its implementation, classification algorithm used, and defining the best 10 and 5 IQMs. In chapter 4, we describe the experiments conducted on RFDS, steps taken for applying these experiments, and the results obtained. Chapter 5 contains the conclusion of the thesis and future work.

## Chapter 2

# ANALYSIS F KNOWN RFDS AND PROBLEM DEFINITION

In this chapter, we analyze some known rfds and list their experimental results. We also list the problems to be analyzed and tested in this thesis.

#### 2.1 Analysis of Known RFDS

# 2.1.1 Image Quality Assessment for Fake Biometric Detection: Application to Iris, Fingerprint, And Face Recognition [6]

In RFDS, they input images and extract their features using 25 IQMs; then, these images will be classified on **Real/ Fake**. The details of how RFDS works are as follows [6]:

1. Training model should be created by feeding in to the system with good amount of real and fake biometric images of users. The images fed to system for creating the training model are known as the training data. The explaniation of how the training model is created is in Chapter 3, section 3.2.3. Once the training model is created, it is exported to the system making the system ready to predict between real and fake images based on the training model.

2. Image enrolment, input some (Real or Fake) gray scale image to the system for identification.

3. Features of the input image will be extracted by the system using the IQM's.

Classification of the input image, the system will classify if the fingerprint image is real or fake image based on the training model exported to it. Steps of the classification implementation are in 3.2.4. Image quality measures in [6].

In RFDS [6], 25 image quality measures have been used split into two parts: 21 of the measures are full reference image quality measurements (image quality measure that requires a reference of similar biometric image type to determine the quality level of the image fed to the system), and 4 are no reference image quality measures (they don't require a reference of similar biometric image type to determine the quality level of an image). Full reference image quality measures to extract the quality features of the original image and enhanced image (original image with adjusted noise is known as enhanced image); no reference image quality measures evaluate the state of the original image inputted to the system. Next we describe the 25 image quality measures used.

#### 2.1.2 Full Reference (FR) Image Quality Measures Used In [6]

**1. Mean Squared Error (MSE):** It calculates the sum of squared difference (error) between the original image (I) and the enhanced image ( $\overline{I}$ ). The mean squared error formula (2.1) [7].

MSE (I. 
$$\overline{I}$$
) =  $\frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (I_{i,j} - \overline{I}_{i,j})^2$ . (2.1)

Where **m**, **n** is the number of columns and rows original image has.

**2. Peak Signal-To-Noise Ratio (PSNR):** It calculates the ratio between the signal power and distortion noise. The peak signal to noise ratio formula (2.2) [8,9].

PSNR (I, 
$$\overline{I}$$
) =10  $log_{10}(\frac{Max(I^2)}{MSE(I,\overline{I})})$ . (2.2)

**3. Signal-To-Noise Ratio (SNR):** Calculates the ratio of power in the input signal to the ration of the noise power, SNR can be defined as the ratio of wanted information to unwanted. Signal-To-Noise Ratio formula (2.3) [10].

SNR (I, 
$$\overline{I}$$
) = 10  $log_{10} \left( \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} (I_{i,j})^2}{N \cdot M \cdot M \cdot M \cdot SE(I,\overline{I})} \right).$  (2.3)

**4. Structural Content (SC):** Calculates the sum of the original image (I) pixel values squared divided by the sum of the enhanced image pixel values ( $\overline{I}$ ) squared. Structural content formula (2.4) [11].

SC 
$$(\mathbf{I}, \bar{\mathbf{I}}) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} (\mathbf{I}_{i,j})^2}{\sum_{i=1}^{N} \sum_{j=1}^{M} (\bar{\mathbf{I}}_{i,j})^2}.$$
 (2.4)

**5. Maximum Difference (MD):** Calculates absolute maximum difference of the original image (I) pixel values subtracted with enhanced image ( $\overline{I}$ ) pixel values. Maximum difference formula (2.5) [11].

$$MD(I,\overline{I}) = Max | \mathbf{I}_{i,j} - \overline{I}_{i,j} |.$$
(2.5)

6. Average Difference (AD): Calculates sum of difference of original image (I) subtracted with enhanced image ( $\overline{I}$ ) divided with total pixels amount the image has. The average difference formula (2.6) [11].

AD (I, 
$$\overline{I}$$
) =  $\frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (I_{i,j} - \overline{I}_{i,j}).$  (2.6)

7. Normalized Absolute Error (NAE): Calculates sum of absolute difference of the original image (I) subtracted with the enhanced image ( $\overline{I}$ ) and divided with sum of absolute pixel values of the original. The normalized absolute error formula (2.7) [11].

NAE (I, 
$$\overline{I}$$
) =  $\frac{\sum_{i=1}^{N} \sum_{j=1}^{M} |I_{i,j} - \overline{I}_{i,j}|}{\sum_{i=1}^{N} \sum_{j=1}^{M} |I_{i,j}|}$ . (2.7)

**8. R**-Averaged Maximum Difference (RAMD): Calculates the maximum difference summation of **R** between the between the original image (I) and the enhanced image (Ī) averaged by **R** value. The R-averaged maximum difference formula [7] (2.8).

RAMD (I, 
$$\bar{I},r$$
)= $\frac{1}{R}\sum_{r=1}^{R} max_r |I_{i,j} - \bar{I}_{i,j}|.$  (2.8)

Where the value of  $\mathbf{R}$  is the number of times we iterate and select the maximum difference between the pixel of the original and enhanced images.

**9. Normalized Cross Correlation (NCC):** Calculates the sum of multiplying the original image (I) and the enhanced image ( $\overline{I}$ ), divided by the sum squared of the original image (I). the normalized cross correlation formula (2.9) [11].

NCC (I, 
$$\overline{I}$$
) =  $\frac{\sum_{i=1}^{N} \sum_{j=1}^{M} (l_{i,j}.\overline{I}_{i,j})}{\sum_{i=1}^{N} \sum_{j=1}^{M} (l_{i,j}.\overline{I}_{i,j})^{2}}$ . (2.9)

10. Total Edge Difference (TED): Calculates the total sum of absolute value of edge difference between the original image (I) pixel values and the enhanced image ( $\overline{I}$ ) pixel values, averaged by total number of pixels in the image. Total edge difference formula (2.10) [12].

TED (I, 
$$\overline{I}$$
) =  $\frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} |I_{Ei,j} - \overline{I}_{Ei,j}|$ . (2.10)

Where  $I_{Ei,j}$  is total number of corners in the original image (I),  $\bar{I}_{ei,j}$  is total number of corners in the enhanced image ( $\bar{I}$ ).

11. Total Corner Difference (TCD): Calculates the total sum of absolute value of corner difference between the original image (I) pixel values and the enhanced image  $(\bar{I})$  pixel values, and then averaged by the maximum number of corners in the image. The total corner difference formula (2.11) [12]

$$\text{TCD}(\mathbf{I}, \overline{\mathbf{I}}) = \frac{|N_{cr} - \overline{N}_{cr}|}{\max(N_{cr}, \overline{N}_{cr})}.$$
(2.11)

Where NCR is total number of corners in the original image (I),  $\overline{N}_{cr}$  is total number of corners in the enhanced image ( $\overline{I}$ ).

12. Spectral Phase Error (SPE): Calculates the sum of phase error between the original image (I) and the enhanced image ( $\overline{I}$ ) squared in the fourier transform and divided by the total number of pixels in the image. Fourier transform transforms original/enhanced image to the fourier domain, where each point represents a particular frequency contained in the image [13]. The spectral phase error formula (2.12) [14].

SPE (I, 
$$\overline{I}$$
) =  $\frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} |\operatorname{Arg}(F_{i,j}) - \arg(\overline{F}_{i,j})|^2$ . (2.12)

Where  $arg(f_{I,j})$  is the phase of the original image, and  $arg(\overline{F}_{i,j})$  is the phase of the enhanced image.

13. Spectral Magnitude Error (SME): Calculates the total sum of absolute value of the difference between the original image (I) and the enhanced image ( $\overline{I}$ ) in the fourier transform averaged by the total number of pixels in the image. The spectral magnitude error formula (2.13) [14].

SME 
$$(I, \bar{I}) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (|F_{i,j}| - |\overline{F}_{i,j}|)^2.$$
 (2.13)

Where  $(|f_i, j| - |\overline{f}_i, j|)$  is the absolute value of difference between two vectors of the original image (I) and enhanced image( $\overline{I}$ ) in the fourier transform.

14. Gradient Phase Error (GPE): Calculates the sum of phase error between the original image (I) subtracted from the original image ( $\overline{I}$ ) squared in the gradient transform and divided by the total number of pixels in the image. Gradient phase error formula (2.14) [15].

GPE 
$$(I, \bar{I}) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} |\operatorname{Arg}(G_{i,j}) - \operatorname{Arg}(\bar{G}_{i,j})|^2.$$
 (2.14)

Where  $arg(g_{I,j})$  is the gradient phase of the original image (i),  $arg(\overline{G}_{I,j})$  is the gradient phase of the enhanced image ( $\overline{I}$ ).

**15. Gradient Magnitude Error (GME):** Calculates the total sum of absolute value of the difference between the gradient of the original image (I) and the gradient of the enhanced image ( $\overline{I}$ ) averaged by the total number of pixels in the image. Gradient magnitude error formula (2.15) [15]:

GME (I, 
$$\overline{I}$$
) =  $\frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (|G_{i,j}| - |\overline{G}_{i,j}|)^2$ . (2.15)

Where **gi**, **j** is the gradient of the original image, and '**gi**, **j** is the gradient of the enhanced image.

**16. Mean Angle Similarity (MAS):** The mean angle that calculates the similarity between the original image (I) and the enhanced image (I). The mean angle similarity formula (2.16) [7]:

MAS (I, 
$$\overline{I}$$
) =  $1 - \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (\alpha_{i,j}).$  (2.16)

Where  $\alpha_{i,j}$  the angle of the original/enhanced image.

**17. Mean Angle Magnitude Similarity (MAMS):** The mean angle calculates the magnitude similarity between the original image (I) and the enhanced image (I). The mean angle magnitude similarity formula (2.17) [7]:

MAMS (I, 
$$\overline{I}$$
) =  $\frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (1 - [1 - \alpha_{i,j}] [1 - \frac{||I_{i,j} - \overline{I}_{i,j}||}{255}]).$  (2.17)

**18. Laplacian Mean Squared Error (LMSE):** Calculates the sum of ratio between the original image(I) and the enhanced image (I) to the original image squared, h(ii, j) = ii + 1, j + ii - 1, j + ii, j + 1 + ii, j - 1 - 4ii. The laplacian mean squared error formula (2.18) [11]:

LMSE (I, 
$$\overline{I}$$
) =  $\frac{\sum_{i=1}^{N-1} \sum_{j=2}^{M-1} (h(I_{i,j}) - h(\overline{I}_{i,j}))^2}{\sum_{i=1}^{N-1} \sum_{j=2}^{M-1} h(I_{i,j})^2}$ . (2.18)

**19. Structural Similarity Measures (SSIM):** It's a quality measure when one single image is contrasted and the other image is still with its original quality, SSIM is an upgrade widespread index.

(Available in [16] and implemented in [17]).

**20. Visual Information Fidelity (VIF):** Considers that the real images are on scenes defined as natural and according to this the type of properties should be the same. (Available in [18] and implemented in [17]).

**21. Reduced Reference Entropy Difference (RRED):** It's a quality measure that extracts some local information of a given sample by processing as a wavelet. Speculation of the sample can't be seen in samples in nature.

(Available in [19] and implemented in [17]).

**22. High Low Frequency Index (HLFI):** Concerned with sharpness and functions by the estimation of the power between the low and up frequency actions of a fourier spectrum.

(Available in [20] and implemented in [17]).

**23. Jpeg Quality Index (JQI):** Evaluation of the distorted image qualities by a known closed artificial initiated comparing algorithm at a decreased number of bit rate like jpeg.

(Available in [21] and implemented in [17]).

**24. Blind Image Quality Index Measures (BIQI):** The basic idea of this measure is that clear original images present some common properties if calculated accurately, divergence of the uniformity in the given statistics in nature is able to estimate the given image quality. BIQI is used for the train image in the past.

(Available in [22] and implemented in [17]).

**25. Natural Image Quality Evaluator (NIQE):** This quality measure evaluates the blind image quality feature when extracting the features of statistics associated to many alterations producing quality information.

(Available in [23] and implemented in [17]).

#### 2.1.3 Classifiers Used in [6] and Our RFDS

Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) are used in [6] to classify fingerprint images if they real or fake.

• Linear Discriminant Analysis (LDA): a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier or, more commonly, for dimensionality reduction before later classification [24].

• Quadratic Discriminant Analysis (QDA): QDA is closely related to linear Discriminant Analysis (LDA), where it is assumed that the measurements are normally distributed. Unlike LDA however, in QDA there is no assumption that the covariance (the measure of how much two or more random variables differ) of each of the classes is identical. To estimate the parameters required in quadratic discrimination more computation and data is required than in the case of linear discrimination [25].

• Linear Support Vector Machine (LSVM): use 3 or 4 support vectors to obtain the optimum boundary between features. For example, we will use 3 support vectors s1, s2, and s3.

$$\hat{S}_{1} = \begin{pmatrix} 2\\1\\1 \end{pmatrix}, \hat{S}_{2} = \begin{pmatrix} 2\\-1\\1 \end{pmatrix}, \text{ And } \hat{S}_{3} = \begin{pmatrix} 4\\0\\1 \end{pmatrix}.$$

Find 3 parameters  $\alpha 1$ ,  $\alpha 2$ , and  $\alpha 3$  based on the following 3 linear equations:

$$\begin{aligned} A_{1}\hat{S}_{1} & \hat{S}_{1} + A_{2}\hat{S}_{2} & \hat{S}_{1} + A_{3}\hat{S}_{3} & \hat{S}_{1} = -1 \ (-negative \ class) \\ A_{1}\hat{S}_{1} & \hat{S}_{2} + A_{2}\hat{S}_{2} & \hat{S}_{2} + A_{3}\hat{S}_{3} & \hat{S}_{2} = -1 \ (-negative \ class) \\ A_{1}\hat{S}_{1} & \hat{S}_{3} + A_{2}\hat{S}_{2} & \hat{S}_{3} + A_{3}\hat{S}_{3} & \hat{S}_{3} = +1 \ (+postive \ class) \end{aligned}$$

Substitute the values for \$1, \$2, and \$3 in the above equations, after simplification of the quations above we get:

$$6\alpha_1 + 4\alpha_2 + 9\alpha_3 = -1$$
  

$$4\alpha_1 + 6\alpha_2 + 9\alpha_3 = -1$$
  

$$9\alpha_1 + 9\alpha_2 + 17\alpha_3 = +1$$

Simplifying the above 3 simultaneous equations we get:

$$\alpha_1 = A_2 = -3.25 \text{ and } \alpha_3 = 3.5.$$

Now we will get the boundary using these three parameters. The boundary between two classes is called hyper  $plane(\hat{w})$ , it discriminates the positive class form the negative class.

$$\hat{\mathbf{w}} = \sum_{i} \alpha_{i} \hat{\mathbf{S}}_{i} .$$
$$\hat{\mathbf{W}} = (-3.25) \cdot \begin{pmatrix} 2\\1\\1 \end{pmatrix} + (-3.25) \cdot \begin{pmatrix} 2\\-1\\1 \end{pmatrix} + (3.5) \cdot \begin{pmatrix} 4\\0\\1 \end{pmatrix} = \begin{pmatrix} 1\\0\\-3 \end{pmatrix} .$$

The vectors are augmented with a bias; we can equate the entry in  $\hat{w}$  as the hyper plane with an offset **b**, therefore the separating hyper plane equation, x and y points on coordinate axis (x1, y1) (x2, y2).

$$Y = wx + b.$$
 (2.19)

 $W = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$  and offset b = -3. This is the expected decision boundary for the linear svm.

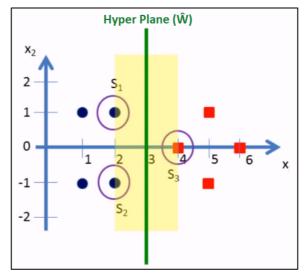


Figure 2: The boundary between two classes (Hyper Plane( $\hat{W}$ )).

• Logistic Regression: Logistic regression predicts whether something is True or False, instead of predicting something continuous like size. Logistic regression doesn't fit a line to the data, instead it fits an "s" shaped "logistic function", where the "s" curve goes from 0 to 1, which means the curve tells us the probability that a fingerprint is Fake or not based on its IQM result.

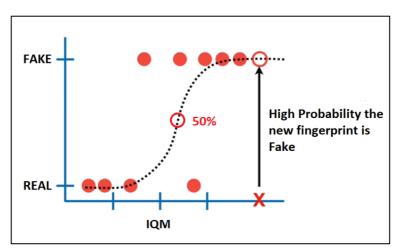


Figure 3: 'S' shaped logistic function.

If the probability a fingerprint is fake is above 50% then we classify it as fake, otherwise we will classify it as real.

• Fine KNN (K-Nearest Neighbors): Fine KNN is a simple way to classify data, if you already had a lot of data that classify the fingerprint, we could use fine KNN to decide which type of class the new fingerprint image(New Cell) belongs to, to be more clear here is an example:

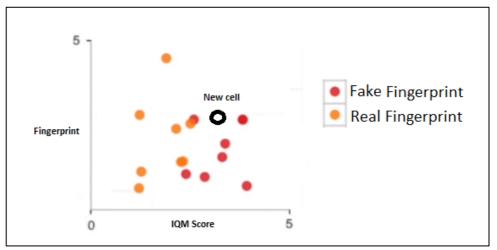


Figure 4: A new cell (fingerprint image) with unknown category to classify.

In figure 4, we added a new cell with unknown category, and we want to classify this new cell. We want to figure out what cell it's most similar to it, then we are going to call it that type of cell. In fine KNN, the new cell will be classified by looking at its nearest neighbors. If "k" in "K-Nearest Neighbors" is equal to 1, then we only use the nearest neighbor to define the category, and in our example the category is red for the new cell because the nearest neighbor is also red.

#### 2.1.3.1 RFDS Experiments In [6]

Experiment was implemented on a system with 64-bit operating system, Windows 7 PC, 3.4 GHz processor, RAM of 16GB, and MATLAB r2012b. LivDet09 [26] was the

database used in the fingerprint detection experiment. LivDet09 [26] has above 18,000 samples obtained from more than 100 different fingers.

Three sensors used in [6] are Biometrika, Crossmatch, and Identix. The overall error rate using biometrika sensor was 12.8%, crossmatch 10.7 %, and identix 1.2%, all shown in Table 1.

**Comparative Results: Fingerprints-LivDet09** Biometrika Crossmatch Identix FGR HTER FFR FFR FFR FGR HTER FGR HTER **IQA-Based** 14.0 11.6 12.8 8.6 12.8 10.7 1.1 1.4 1.2

Table 1: RFDS [6] results using three different type of fingerprint sensors.

Description of the measures used in Table 1:

False Fake Rate (FFR) refers to the number of real images that were considered as fake in the system.

$$FFR = \frac{FFR}{TOTAL NUMBER OF REAL IMAGES}.$$
 (2.20)

False genuine rate (FGR) refers to number of fake images that were considered as real in the system:

$$FGR = \frac{FGR}{TOTAL NUMBER OF FAKE IMAGES} .$$
 (2.21)

Half Total Error (HTER) refers to the number of real images that were considered as fake in the system summed with the number of fake images that were considered as real in the system divided by 2,

$$HTER = \frac{FGR + FFR}{2}.$$
 (2.22)

#### 2.1.4 Fingerprint Liveness Detection Based on Quality Measures [27]

In [27], the RFDS uses image quality measures for a Software-Based solution in fingerprint image detection. The main purpose of [27] is to counter the direct attacks implemented against the fingerprint recognition systems.

Direct attacking methods present artificial fingerprints to the sensors so that the sensor will identify them as real fingerprints inserted to the system, and thus access is granted. To prevent direct attacking occurrence, a new live detection system was proposed in this paper with 3 types of image quality measures, ridge strength, ridge continuity, and ridge clarity. The new live detection system proposed in [27] includes the steps that are explained below:

**Input:** Once the fingerprint image gets inserted into the system, it will be immediately segmented (Image is Easier to Analyze) in the background.

**Feature Extraction:** Image quality measures will be extracted, the image quality measures with the best performance only will be selected from the all the available ones. After selecting the best image quality measures, we reach the final stage, cclassification of the fingerprint image. Linear discriminant analysis (LDA) (described in subsection 2.1.1) is used in [27] to classify whether the inserted fingerprint image is Real or Fake.

#### 2.1.4.1 RFDS Experiments In [27]

Database used in [27] is provided in the fingerprint liveness detection competition, livdet 2009, containing over 18,000 real and fake samples of fingerprint images generated with different materials and captured with different sensors. Three sensors were used in [27]: biometrika, crossmatch, and identix.

The proposed new liveness detection system result was on average (7.58% hter) in identifying real or fake sample of fingerprint images. The results of [27] for each sensor are shown in Table 2.

	Fgr(%)	Ffr(%)	Hter(%)
Bionetrika	2.12	1.54	1.83
Crossmatch	12.48	12.32	12.40
Identix	6.40	10.67	8.53
Average	7.00	8.17	7.58

Table 2: The results for each sensor [27].

# 2.1.5 A High-Performance Fingerprint Liveness Detection Method Based on Quality Related Features [28]

A software-based fingerprint liveness detection approach using image quality measures is proposed in [28]. The system was tested by using two different databases and five different sensors.

The proposed system starts with inserting a fingerprint image and, the type of sensor used for getting the fingerprint image. After inserting the sensor type and the fingerprint image, segmentation will be applied to the fingerprint image, and then three different types of image quality measures will be extracted from the fingerprint image: ridge strength, ridge continuity, and ridge clarity.

After applying extraction to the fingerprint image and obtaining the results of every single image quality measurement, selection of the ones with the best results from the image quality measures will be done, and the rest will be excluded for the classification

of the image stage. Classification is done using LDA (defined in 2.1.1) to classify if the fingerprint image is real or fake.

## 2.1.5.1 RFDS Experiments In [28]

In [28], two databases were used: LivDet DB and ATVS DB. ATVS database contains three different datasets of real and fake fingerprints captured each by three different sensors: Biometrika, Precise, and thermal Yubee. LivDet database contains also three different datasets of real and fake fingerprints captured each captured by three different sensors, Namely, Biometrika, Crossmatch, and Identix.

Experiments results in [28] with both ATVS DB and LivDet show overall around 10% of error in identifying real and fake fingerprint images in the liveness detection system. The results for both databases with five different sensors are listed in Table 3, FGR<sub>1</sub>, FFR<sub>1</sub>, and HTER<sub>1</sub> REFER to the ATVS db, and FGR<sub>2</sub>, FFR<sub>2</sub>, and HTER<sub>2</sub> refer to the LivDet.

	Performance Results In %						
	FGR <sub>1</sub> /FGR <sub>2</sub>	FFR <sub>1</sub> /FFR <sub>2</sub>	HTER <sub>1</sub> /HTER <sub>2</sub>	Hter			
Biomet.Ld	3.1/9.8	71.8/21.5	37.4/15.6	26.5			
Crossmatch	8.8/16.7	20.8/6.8	14.8/11.7	13.2			
Identix	4.8/8.0	5.0/9.1	4.9/8.5	6.7			
Biomet.Atvs	9.4/0.4	1.5/11.8	5.5/6.1	5.8			
Precise	3.1/0.4	13.7/0.4	8.4/0.4	4.4			
Yubee	2.7/1.6	6.3/13.0	4.5/7.3	5.9			
Average	5.3/6.1	19.8/10.4	12.5/8.2	10.4			

Table 3: Performance result of the two databases used in [28].

# 2.1.6 Fingerprint Identification in Biometric Security Systems. International Journal of Computer and Electrical Engineering [29]

In [29], two software based matching algorithms (algorithms that match fingerprints against a database of known and unknown fingerprint) are discussed. The first algorithm is developed at the Hong Kong Baptist University; the second matching algorithm is developed at the michigan state university in usa. Two of the algorithms were downloaded and tested with a common database.

The Hong Kong Baptist University fingerprint identification system uses a traditional minutia matching algorithm, while the Michigan State University uses filter-based algorithm in their system.

The Hong Kong baptist university fingerprint identification system is based on three steps:

**1. Preprocessing:** The inserted fingerprint image is enhanced using fourier transform (conversion of the image from space domain to frequency domain) and histogram equalization (modify the intensity and contrast of the fingerprint image), after that binarization is applied to the image which will change it from pixel image to a binary image using threshold for each block in the image.

**2. Minutiae Extraction:** The extraction is done by thinning the inserted fingerprint image and marking any minutiae that exists in the image.

**3. Post-Processing:** In this stage, all false minutiae (major features of a fingerprint such as ridge ending and bifurcation) are removed.

Michigan State University method uses filter-bank based matcher: the proposed filterbank based algorithm uses more than one filter to capture details in a fingerprint as a compact fixed length feature vector. The fingerprint matching is based on the euclidean distance between the two corresponding vectors.

#### 2.1.6.1 RFDS Experiments In [29]

The database used in [29] is NIST special database 4 which is publicly available; it contains 8-bit gray scale image fingerprints distributed for development and testing of fingerprint identification systems.

For 25 pairs of fingerprints with high quality inserted into the software using filterbased algorithm, the results are as follows: 8% FAR and 4% FFR.

In [29], they concluded that both of the algorithms are equal in performance and none of them can be considered as better than the other in general sense.

#### 2.1.7 Fingerprint Spoof Detection Using Quality Features [30]

In [30], three stages are considered for fake fingerprint detection:

**1. Feature Extraction:** The inserted fingerprint image is divided into blocks and the features are extracted from each block. There is a total of 7 features which will be get extracted from the input fingerprints image.

**2. Features Selection:** The features with optimal performance will be selected; the ones that are not effective, will be excluded.

**3.** Classification: After feature extraction and selecting the best extracted features, classification of the fingerprint image is implemented to identify if the inserted fingerprint image is real or fake.

#### 2.1.7.1 RFDS Experiments In [30]

The database used in [30] is provided in the fingerprint liveness detection competition LivDet 2009 [26] with three data sets of real and fake fingerprint images all captured with different kind of optical sensors: Biometrika, Identix, and Crossmatch. Experiment results in [30] are considered overall as a good RFDS to use, the results in Figure 5 show that if all features are used at once without excluding any one of them, better results will be obtained.

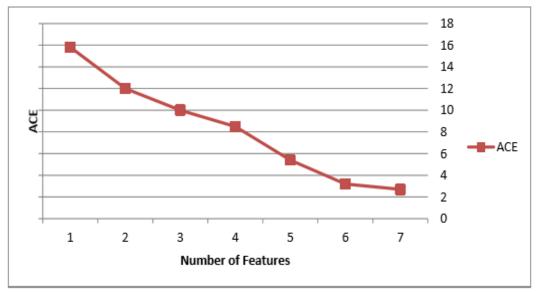


Figure 5: Average classification error (ACE) results in [30].

# 2.2 Problem Definition

We have observed [6], [18], [19], [20], and [30], but we will mainly compare our RFDS to be developed with RFDS [6]. The list of problems considered in this thesis is:

- 1. Implement and test a real fingerprint detection system (RFDS) similar to [6].
- 2. Repeat experiments on RFDS implemented as in [6].
- 3. Compare our RFDS with [6].
- 4. Find the best 5, 10, and 15 iqms in our RFDS.

5. Modifying thesis [31] the distinction from our RFDS and thesis [31] is that we used 10 more IQMs and 6 classifiers. Thesis [31] was implemented for face detection while ours is for fingerprints.

# 2.3 Summary

In this chapter, we have done a literature review from the reading and analysis of [6], [27], [28], [29], and [30]. We conclude that some of the presented RFDS use similar image quality features and classification algorithms and some use different. The experiment results on these systems were gathered using different databases, all these results obtained still didn't reach 100% accuracy of detecting fingerprint spoof attacks. The features to be built in our RFDS is to find the best 5, 10 and 15 IQM's automatically calculating the FFR, FAR, and HTER of the RFDS. We define our thesis problem as: implementation and testing of real fingerprint image detection system (RFDS).

# Chapter 3

# **DESIGN AND IMPLEMENTATION OF RFDS**

In this chapter, we show the design of our RFDS, implementation of the feature extraction in our system as well as the classification process. We implemented the Gaussian filter as explaned in subsection 3.2.1, feature extraction stage for the fingerprint images as explaned in subsection 3.2.2 using 25 image quality measures. The training structure was implemented and explaned in subsection 3.2.3, classification process in subsection 3.2.4, method of defining the best IQMs in subsection 3.2.5, and the method of defining the best IQMs was implemented and tested as described in subsection 3.2.

## 3.1 Design of RFDS

Figure 6 shows the design of our RFDS. Basically, our system contains two structures:

- 1. Training Structure (as shown at the top of figure 6)
- 2. Detection Structure (as shown at the top of figure 6).

The top of Figure 6 shows our RFDS training structure where (I) refers to the original fingerprint image and ('I) refers to the enhanced image. Images will be input in our RFDS training structure. In block 1, each single image will be filtered using gaussian filter (see subsection 3.2.1) with 3\*3 kernel to reduce the noise of the image. By filtering the input images (I), we get the enhanced fingerprint image (I'). In block 2, we will apply feature extraction process to both of the fingerprint images by extracting image quality measures (see subsection 3.2.2), we then create a table and list the results

of the fingerprint images and their IQM. In block 3, we import the table with all the results to classification learner application, which is a user-friendly application that we use in MATLAB software to create our training model (block number 4) and classify images (see subsection 3.2.3). From this we move to the classification stage where all the fingerprint images will be classified as either real or fake images.

The bottom of Figure 6 shows our RFDS detection structure. All input images will be classified on real or fake depending on the trained model created in the "training structure" shown in figure 6.

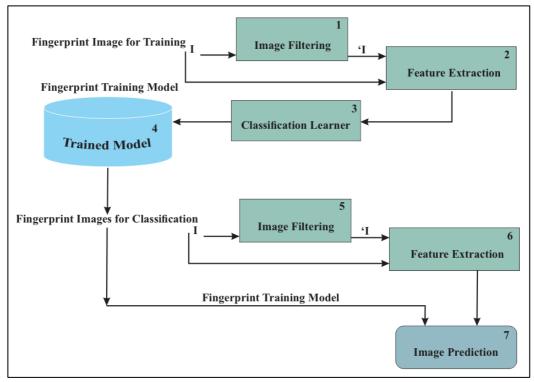


Figure 6: RFDS training and detection structure.

## **3.2 Implementation of RFDS**

#### 3.2.1 Implementation and Testing of Gaussian Filter

Gaussian filter is a filter whose impulse response is a Gaussian function (or an approximation to it). Gaussian filters have the properties of no overshoot to a step

function input while minimizing the rise and fall time [32]. For our RFDS implementation and testing, we used MATLAB 2015 in this thesis. For implementation of Gaussian filter, we used a ready MATLAB function "Gaussian" to enhance and reduce the noise of the image. All we had to do was to call the function by typing "Gaussian" which is shown below in Figure 7, and highlighted with number 1. Then, we set the value of the Gaussian kernel as "3x3" highlighted and numbered 2 in the screenshot, and set the standard deviation as "0.6" (0.6 was selected just as an example) highlieghted and numbered 3 in Figure 7. By these three values, we set our RFDS system that gives us a Gaussian filter with the values shown in number 4 of the screenshot below.

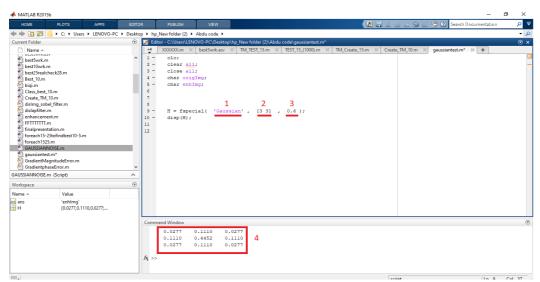


Figure 7: Gaussian filter code and the Gaussian filter obtained from it.

Example of how Gaussian filter is implemented using a 4\*4 matrix image:

Original Image					
0	1	7	3		
4	2	8	10		
0	2	1	6		
6	3	5	4		

3x3 Gaussian Kernel					
0.0277	0.1110	0.0277			
0.1110	0.4452	0.1110			
0.277	0.1110	0.0277			

Let's select the pixel in 2nd row 3rd column of the original image and apply Gaussian filter to it:

(	0	1	7	3		0.0277	0.111	0.0277		0.0277	0.777	0.0831
4	4	2	8	10	*	0.1110	0.4452	0.111	=	0.222	3.5616	1.11
(	0	2	1	6		0.0277	0.111	0.0277		0.0554	0.111	0.1662
(	6	3	5	4			1					

Add the values in the vector: 0.0277 + 0.777 + 0.0831 + 0.222 + 3.5616 + 1.11 + 0.0554 + 0.111 + 0.1662 = 6.114

0.610	1.776	4.780	3.444
2.085	2.777	6.114	6.560
1.470	2.193	3.302	4.696
3.059	2.806	3.335	3.029

Final result of the original image after Gaussian filter as.

To check if our manual calculation of the Gaussian filter applied to the original image is correct or not, we ran our MATLAB code and took a screenshot of the result shown below in figure 8, the result of the original image with Gaussian filter applied to it is the same as (3.1).

📣 MATLAB R2015b	- Ø	×
HOME PLOTS APPS	EDITOR PUBLISH VEW	٩
	To Comment % % % % % % % % % % % % % % % % % % %	•
AMD4:m  Addum   <pre>2 - clc; 3 - clear all; 4 - close all; 5 - char origimg; 6 - char distImg; 7 - origimg = [0,1,7,3;4,2,8,10;0,2,1,6;6,3,5,4]; 9</pre>		
Workspace 💿		
Name ▲         Value           ans         'distImg'           enhimg         4x4 double           GaussF         4x4 double	Command Window New to MATLAB? See resources for <u>Getting Started</u> .	(
H [0.0277,0.1110,0.0277; origimg 4x4 double R 4x4 double	0.6104 1.7765 4.7809 3.4443 2.0859 2.7771 <mark>6.1145</mark> 6.5608 1.4706 2.1932 3.3025 4.6964 3.0597 2.8065 3.3356 3.0297	
< <u> </u>	<u>R</u> >	

Figure 8: Result of the original image with Gaussian filter applied to it.

## **3.2.2 Implementation and Testing of Feature Extraction**

To test the implementation of our RFDS, we use 4x4-sized (for ease of calculation) two gray scale images, one named original image (I) with matrix elements: (0, 1, 7, 3),

(4, 2, 8, 10), (0, 2, 1, 6), (6, 3, 5, 4) and the second one named enhanced image ('I) (original image with Gaussian filter applied to it) with matrix elements: (2, 5, 10, 4), (0, 1, 5, 1), (3, 6, 2, 3), (1, 3, 10, 0).

Original Image (I)					
0	1	7	3		
4	2	8	10		
0	2	1	6		
6	3	5	4		

Enhanced Image ('I)					
2	5	10	4		
0	1	5	1		
3	6	2	3		
1	3	10	0		

For each of the 25 IQMs we will do the following (our MATLAB code is provided in Appendix A and the IQM function in Appendix B).

For 4x4-sized of the enhanced and original image, we will take screenshots of every IQM result in MATLAB.

## 3.2.2.1 Implementation and Testing of MSE (2.1)

Explanation of MSE code given in (Appendix A-1):

Line 1: MSE function has two inputs: rel\_img refers to the real image, and enhcd\_img to the enhanced image. Line 4: [m n] is defined where m is the number of rows and n is the number of columns the real image has. Line 5: difference between real and enhanced images is calculated. Line 6: using (2.1), MSE is calculated.

Testing example of MSE calculation for I and 'I:

$$\mathbf{MSE} = \frac{1}{16} * ((0-2))^{2} + (1-5))^{2} + (7-10))^{2} + (3-4))^{2} + (4-0))^{2} + (2-1))^{2} + (8-5))^{2} + (10-1))^{2} + (0-3))^{2} + (2-6))^{2} + (1-2))^{2} + (6-3))^{2} + (6-1))^{2} + (3-3))^{2} + (5-10))^{2} + (4-0))^{2} = \frac{1}{16} * (238) = 14.875$$
(3.2)

The result of MSE calculation shown in Figure 9 is equal to our manually calculated result (3.2).

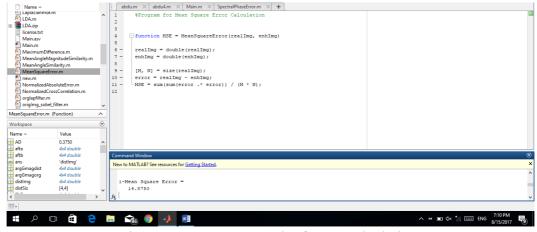


Figure 9: MATLAB result of MSE calculation.

## 3.2.2.2 Implementation and Testing of PSNR (2.2)

Explanation of the PSNR code (Appendix A-2):

Line 1: PSNR function with two inputs rel\_img and enhcd\_img refers to the real and enhanced image, respectively. Line 4: [m n] is defined, where m and is the number of rows and columns the real image has. Line 5: difference between real image and enhanced image will be calculated. Line 6: using (2.1), MSE is calculated. Line 7: using (2.2), PSNR is calculated. Line 8: we set PSNR value as 99 if MSE is zero.

MSE = 14.875 (Obtained from Figure 9).

Max \_I = 255 (max pixel value in grayscale image).

 $PSNR = 20log_{10}255 - 10log_{10}MSE$  $= 20log_{10}255 - 10log_{10}14.875 = 48.13 - 11.72 = 36.40$ (3.3)

The result of PSNR calculation shown in Figure 10 is equal to our manually calculated result (3.3).

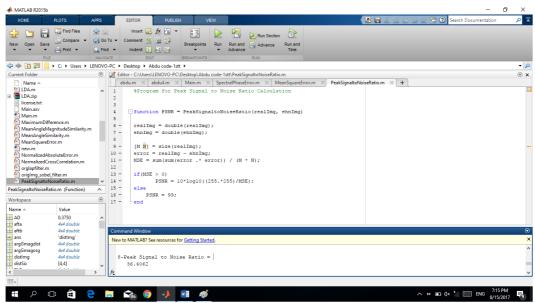


Figure 10: MATLAB result of PSNR calculation.

# 3.2.2.3 Implementation and Testing of SNR (2.3)

Explanation of the SNR code (Appendix A-3):

Line 1: SNR function with two inputs: rel\_img and enhcd\_img refer to the real and enhanced image, respectively. Line 4: [m n] is defined, where m and n are the number of rows and columns of the real image. Line 5: difference between real image and enhanced image is calculated. Line 6: using (2.1), MSE is calculated., Line 7: using (2.3), SNR (2.3) is calculated. Line 8: we set SNR to 99 if MSE is zero.

**MSE** = 14.875 (obtained from Figure 9)

SNR Calculation:

 $SNR = 10 \text{ Log}10 (0^{2} + 1^{2} + 7^{2} + 3^{2} + 4^{2} + 2^{2} + 8^{2} + 10^{2} + 0^{2} + 2^{2} + 1^{2} + 6^{2} + 6^{2} + 3^{2} + 5^{2} + 4^{2}) / 4 * 4 * 14.875$ SNR = 10 Log10 370/238 = 10 Log10 1.55 = 1.91(3.4) The result of SNR calculation shown in Figure 11, is equal to our manually calculated result (3.4).

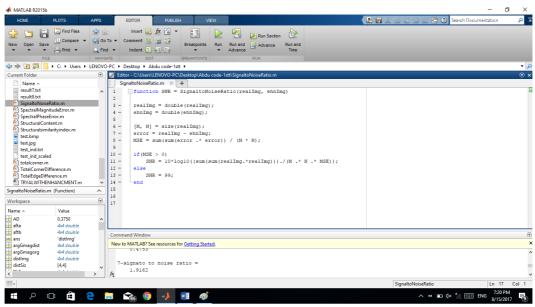


Figure 11: MATLAB result of SNR calculation.

# 3.2.2.4 Implementation and Testing of SC (2.4)

Explanation of the SC code (Appendix A-4):

Line 1: SC function with two inputs rel\_img and enhcd\_img refers to the real and enhanced image. Line 4: using (2.4), SC is calculated.

$$SC = (0^{2} + 1^{2} + 7^{2} + 3^{2} + 4^{2} + 2^{2} + 8^{2} + 10^{2} + 0^{2} + 2^{2} + 1^{2} + 6^{2} + 6^{2} + 3^{2} + 5^{2} + 4^{2}) / (2^{2} + 5^{2} + 10^{2} + 4^{2} + 0^{2} + 1^{2} + 5^{2} + 1^{2} + 3^{2} + 6^{2})$$

$$+2^{2}+3^{2}+1^{2}+3^{2}+10^{2}+0^{2}=370/340=1.0882.$$
(3.5)

The result of SC calculation shown in Figure 12, is equal to our manually calculated result (3.5).

📣 MATLAB R2015b		o ×
HOME PLOTS APPS	EDITOR PUBLISH VIEW Search Documentation	<mark></mark> 🔎 🔺
New Open Save	Insert       Ar       Ar       Image: Section for the section for th	
	VU-PC > Desktop > Abdu code-1stt >	<mark>م ד</mark> ج ج
Name A	StructureContention - Closekup Gala coder siccinetium	• •
result&at SignitubNoce#atic.m SpectralMagnitudeForc.m SpectralMagnitudeForc.m SpectralMagnitudeForc.m StructuralGometer.m StructuralGometer.m test_ind_ test_ind_staled totalCometifierence.m TotalEgaPdEference.m		
Name 🔺 Value		
AD 0.3750 afta 4x4 double aftb 4x4 double	Command Window	$\overline{\mathbf{O}}$
ans 'disting'		
argGmagdist 4x4 double argGmagorg 4x4 double distImg 4x4 double distSiz [4,4]	New to MATLAB? See resources for <u>Getting Statted</u> . 14-Structural Content = 1.0882 Æ	-
	🛤 💿 🛃 👰 лин ю фх % 📰 выс 221 г уллаг	

Figure 12: MATLAB result of SC calculation.

# 3.2.2.5 Implementation and Testing of MD (2.5)

Explanation of MD code (Appendix A-5):

Line 1: MD function with two inputs: rel\_img and enhcd\_img refer to the real and enhanced image. Line 4: the difference between the real image and the enhanced image will be calculated. Line 5: using (2.5), MD will be calculated.

P1=0-2=-2		P9 = 0 - 3 = -3	
P2 = 1 - 5 = -4		P10 = 2 - 6 = -4	
P3 = 7 - 10 = -3		P11 = 1 - 2 = -1	
P4 = 3 - 4 = -1		P12 = 6 - 3 = 3	
P5 = 4 - 0 = 4		P13 = 6 - 1 = 5	(3.6)
P6 = 2 - 1 = 1		P14 = 3 - 3 = 0	
P7 = 8 - 5 = 3		P15 = 5 - 10 = -5	
P8 = 10 - 1 = 9		P16=4-0=4	
	Md = 9		

The result of MD calculation shown in Figure 13 is equal to our manually calculated result (3.6).

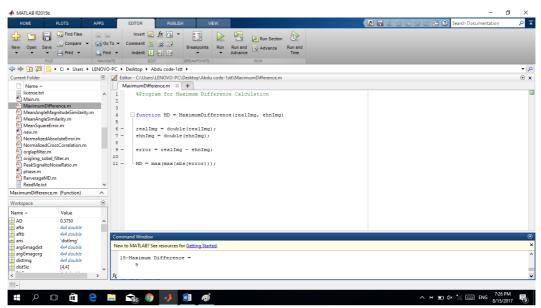


Figure 13: MATLAB result of MD calculation.

# 3.2.2.6 Implementation and Testing of AD (2.6)

Explanation of the AD code (Appendix A-6):

Line 1: AD function with two inputs: rel\_img and enhcd\_img refer to the real and enhanced image, line 4: [m n] is defined, where m and n is the number of rows and columns of the real image. 5: difference between real image and enhanced image will be calculated. Line 6: using (2.6), ad will be calculated.

$$\mathbf{AD} = \frac{1}{16}((0-2) + (1-5) + (7-10) + (3-4) + (4-0) + (2-1) + (8-5) + (10-1) + (0-3) + (2-6) + (1-2) + (6-3) + (6-1) + (3-3) + (5-10) + (4-0)) = \frac{1}{16}(-2-4-3-1+4+1+3+9-3-4-1+3+5+0-5+4) = \frac{6}{16} = 0.375$$
(3.7)

The result of ad calculation shown in Figure 14 is equal to our manually calculatedresult (3.7).

HOME       RADIS       EDTOR       NULLEN       VEV       Image: Comment of the second	×
Norme ^       VersepOrterencem       1       VersepOrterence       VersepOrterence       VersepOrterence         Norme ^       Voltage       0       0.0000       0.0000       0.0000       0.0000         Norme ^       VersepOrterence       VersepOrterence       VersepOrterence       VersepOrterence       VersepOrterence         Norme ^       Voltage       0.0000       0.0000       0.0000       0.0000       0.0000         Norme ^       VersepOrterence       VersepOrterence       VersepOrterence       VersepOrterence       VersepOrterence         Norme ^       Voltage       0.0000       0.0000       0.0000       0.0000       0.0000         Norme ^       Voltage       0.0000       0.0000       0.0000       0.0000       0.0000         Norme ^       Voltage       0.0000       0.0000       0.0000       0.0000       0.0000         VersepOrterence       VersepOrt	<mark>م</mark>
Current Folder       Ø       Eddor = Childron JEDVO -PC Desktop Abdu code-1stt AverageDifferencem         A mere a       AverageDifferencem       A +         B abdu       AverageDifference Calculation       Image: Calculation         B abdu       Image: Calculation       Image: Calculation         Image: Calculation       Image: Calculation       Image: Calculation         Image: Calculatine       Image: Calculatine	
Name *       Value         Inspectation       Image dabase         Average/Interaccen       Image dabase         AbdUm       Image dabase         Image dabase       Image dabase         Image dabase       Image dabase         AbdUm       Image dabase         Image dabase       Image dabase         AbdUm       Image dabase         Abdum       Image dabase         Image dabase       Image dabase         Abdum       Image dabase         Abdum       Image dabase         Abdum       Image dabase         Image dabase       Image dabase         Abdum       Image dabase         Image dabase       Image dabase <t< td=""><td>- 2</td></t<>	- 2
addu       %Program for Average Difference Calculation         finge database       %Program for Average Difference Calculation         addum       addum         abdum       % addum         bdum       % addum         isduding       % addum         isduding obde filter.m       %         % enhancement.m       %         % AverageDifference function          % Addispliker.m          % addispliker.m </td <td>0 :</td>	0 :
Image database       33ADAm         33ADAm       33ADAm         33ADAm       34dus         34dus       34dus         34dus	
aftb 4wl double Command Window	
	(
argGmagdiat     dud Gouble       argGmagory     dud Gouble       dudting     dud double       0.3750	
	Col 1
📲 Р 🗇 🛱 🤤 🔚 🟫 🧔 🛷 💵 🛷	

Figure 14: MATLAB result of AD calculation.

## 3.2.2.7 Implementation and Testing of NAE (2.7)

Explanation of NAE code (Appendix A7):

Line 1: NAE function with two inputs: rel\_img and enhcd\_img refer to the real and enhanced image. Line 4: the difference between the real and enhanced image will be calculated. Line 5: using (2.7), NAE will be calculated.

$$\mathbf{NAE} = [(0-2) + (1-5) + (7-10) + (3-4) + (4-0) + (2-1) + (8-5) + (10-1) + (0-3) + (2-6) + (1-2) + (6-3) + (6-1) + (3-3) + (5-10) + (4-0)]$$
$$\mathbf{Nae} = [2+4+3+1+4+1+3+9+3+4+1+3+5+0+5+4] = 52/62 = 0.8387$$
(3.8)

The result of NAE calculation shown in figure 15 is equal to our manually calculated result (3.8).

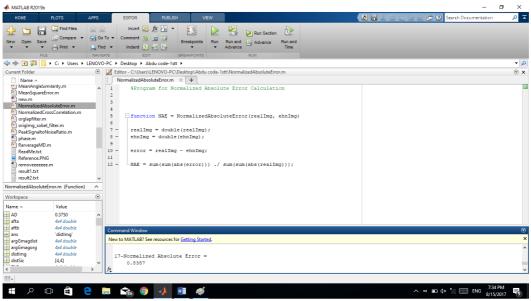


Figure 15: MATLAB result of NAE calculation.

# 3.2.2.8 Implementation and Testing of RAMD (2.8)

Explanation of the RAMD code (Appendix A-8):

Line 1: RAMD function with two inputs: rel\_img and enhcd\_img refer to the real and enhanced image. Line 4: the difference between the real and enhanced image is calculated. Line 5: the variable 'eror' will have the absolute value of error. Line 6: we convert matrix to vector, then we order the numbers by descending order (using flipud a ready MATALAB function) without having repeated numbers, all numbers should be unique. Line 7: select top ten from the vector. Line 8: using (2.8), RAMD is calculated.

R = 7 (we select the maximum 7 differences between the image I and 'I and divide over 7, in the main code r=10, but for the testing case we set R=7 because if we set R=10 the 8<sup>th</sup>, 9th, and 10<sup>th</sup> values will be equal to zero).

P1 = 0 - 2 = -2	P9 = 0 - 3 = -3	
P2 = 1 - 5 = -4	P10 = 2 - 6 = -4	
P3 = 7 - 10 = -3	P11 = 1 - 2 = -1	
P4 = 3 - 4 = -1	P12 = 6 - 3 = 3	(3.9)
P5 = 4 - 0 = 4	P13 = 6 - 1 = 5	
P6 = 2 - 1 = 1	P14 = 3 - 3 = 0	
P7 = 8 - 5 = 3	P15 = 5 - 10 = -5	
P8 = 10 - 1 = 9	P16=4-0=4	
<b>RAMD</b> = $1/7   9 + 5 + 4 + 3 + 2 + 3 + 2 + 3 + 2 + 3 + 2 + 3 + 3$	1 = 24/7 = 3.428	

The result of RAMD calculation shown in Figure 16 is equal to our manually calculated result (3.9).

📣 MATLAB R2015b		- a ×
HOME PLOTS	APPS EDITOR PUBLISH VIEW 🚺 🖬 🔬 🖄 🖄 🗇 🗇 🗇 🕐 Search Docum	nentation 🔎
New Open Save	Insert     fr     fr     Image: Comment     fr	
🗆 🔶 🔁 🔀 📙 🕨 C: 🕨 Users 🕨	LENOVO-PC + Desktop + Abdu code-1stt +	•
Current Folder	🕑 💋 Editor - C:\Users\LENOVO-PC\Desktop\Abdu code-1stt\RarverageMD.m	$\odot$
Name A	RarverageMD.m 🕱 +	
NormalizedAbsoluteTrocm     NormalizedAbsoluteTrocm	<pre>     function RAMD = RarverageMD(realImg, ehnImg)     f     f         fe = realImg = double(realImg);         fe = realImg = double(tehnImg);         fe = rearrow(tehnImg);         fe = rearrow(tehnImg);</pre>	~
aftb 4x4 double ans 'distImg'	Command Window	
ans disting argGmagdist 4x4 double argGmagorg 4x4 double distmg 4x4 double distSiz [4,4]	New to MATLA87 See resources for <u>Gatting Stanted</u> . 16-RaverageMD = 3.4286 //	
<b>*</b>	RarverageMD	Ln 11 Col 3
ים א 🖬	😫 🔚 😪 🌖 🛃 🍯	NG 8/15/2017

Figure 16: MATLAB result of RAMD calculation with R=7.

# 3.2.2.9 Implementation and Testing of NCC (2.9)

Explanation of NCC code (Appendix A-9):

Line 1: NCC function with two inputs: rel\_img and enhcd\_img refer to the real and enhanced image. Line 4: using (2.9) NCC will be calculated.

$$NCC = ((0*2) + (1*5) + (7*10) + (3*4) + (4*0) + (2*1) + (8*5) + (10*1) + (0*3) + (2*6) + (1*2) + (6*3) + (6*1) + (3*3) + (5*10) + (4*0) / (0^2 + 1^2 + 7^2 + 3^2 + 4^2) + 2^2 + 8^2 + 10^2 + 0^2 + 2^2 + 1^2 + 6^2 + 6^2 + 3^2 + 5^2 + 4^2) = NCC = 236 / 370 = 0.637$$

$$(3.10)$$

The result of NCC calculation shown in Figure 17 is equal to our manually calculated result (3.10).

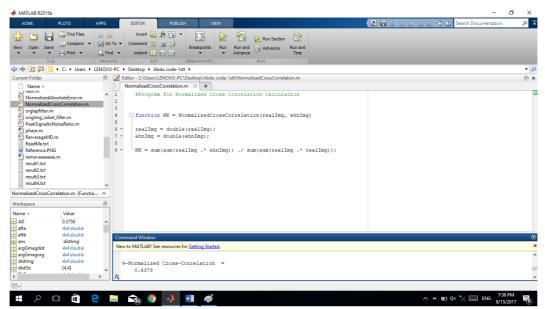


Figure 17: MATLAB result of NCC calculation.

#### 3.2.2.10 Implementation and Testing of TED (2.10)

Explanation of TED code (Appendix A10):

Line 1: TED function with two inputs: rel\_img and enhcd\_img refer to the real and enhanced image. In Line 2, ready MATLAB function (edge) is used to detect the edges of the real image. In Line 3, all edges that were detected by the MATLAB function will be displayed. In Line 5 and 6 the same process is done as in lines 2 and 3 but this time for the enhanced image. Line 8: [m n] is defined, where m is the number of rows and n is the number of columns the real image has. Line 9: difference between real image and enhanced image will be calculated and saved in 'eror' variable. Line 10: using the formula provided in (2.10) TED will be obtained.

We didn't apply manual calculation to test TED because in our code we use ready MATLAB function "edge (image)" to automatically find the image total edge difference of any image inserted to the RFDS, so we feed a fingerprint image to our system and took a screenshot of the result shown below in figure 18.

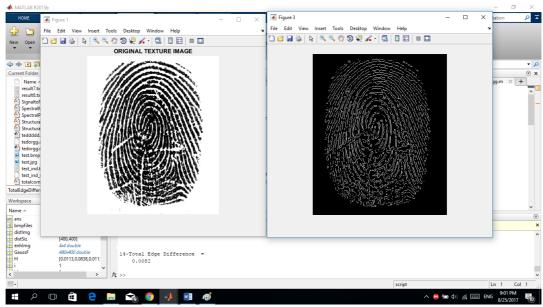


Figure 18: Matlab result for calculating TED of fingerprint image.

## **3.2.2.11** Implementation and Testing of TCD (2.11)

Explanation of the TCD code (each line number corresponds to its code in Appendix A-15):

Line 1: TCD function with two inputs rel\_img which refers to the real image and enhcd\_img refers to the enhanced image, line 4 we use a ready MATLAB function called "detectharrisfeatures" to detect corners of the real image, Line 5: we plot the corners in the real image using a MATLAB function called "plot", Line 6: we will get the sum of corners of the real image using ready MATLAB function called "length", Line 7 we will detect corners of the enhanced image, Line 8: we plot the corners of the enhanced image same as we did to the real image, Line 9: we will get the sum of corners in of the enhanced image using the same function we applied to the real image, Line 10: we will set the value of variable "max1" either as the number of corners of the real or enhanced image depending on which one has th maximum number of corners detected, Line11: using the formula provided in (2.11) TCD will be calculated. **Ncr** = Number of Corners in The Original Image.

**'Ncr** = Number of Corners in The Enhanced Image.

Ncr= 2496

**'Ncr**= 2400

 $\mathbf{Tcd} = |2496-2400| / 2496 = 0.0385 \tag{3.11}$ 

The result of TCD calculation shown in Figure 19 is equal to (3.11).

📣 MATLAB R2015b		- 0 ×
HOME PLOTS APPS	EDITOR PUBLISH VEW	📕 🗒 🔏 🕼 🚔 🗇 🗭 🔁 Search Documentation 🖉
Image: Save     Image: Find Files     Image: Save     Image: Save <t< th=""><th>Insert     fx     fx     Image: Section     Image: Section<!--</th--><th></th></th></t<>	Insert     fx     fx     Image: Section     Image: Section </th <th></th>	
Marre -     SpectalPacEnorm     SpectalPacEnorm     StructuralContentum	<pre>3ANDAm X enhancementm X ffTTTTTTT X hhhm X news X 1</pre>	Main.m X TotalEdgeDifference.m X tedorggm X TotalCornerDifference.m* X +
bmpfiles 50x1 struct distImg 480x400 double distSiz [480,400] enhImg 4x4 double	20 Command Window New to MATLAB? See resources for Getting Started.	· (
GaussF 480x400 double H [0.0113,0.0838,0.011: i 1	>> 15-TotalCornerDifference =	
		TotalCornerDifference Ln 12 Col 40

Figure 19: MATLAB result of TCD calculation.

3.2.2.12	Implementation	and Testing	of Spectral	<b>Phase Error</b>	(SPE)
					( - )

Original Image				
0	1	7	3	
4	2	8	10	
0	2	1	6	
6	3	5	4	

Explanation of the SPE code (each line number corresponds to its code in Appendix B-14):

Line 1: SPE function with two inputs rel\_img which refers to the original image and enhcd\_img refers to the enhanced image, Line 4: we use a ready MATLAB function "fft" to transform the original and enhanced image to the frequency domain, Line 5: [m n] where m is amount of row the real image has and n is amount of columns the real image has; Line 6: difference between real image and enhanced image will be calculated, Line 7: using the formula provided in (2.12) spe will be calculated.

FFT Original Image				
62 +0i	-11+15i	0+0i	-11-	
2-6i	1+3i	10+4	-13+7i	
-22+0i	-5-3i	-8+0i	-5+3i	
2+6i	-13-7i	10-4i	1-3i	

Distorted Image				
2	5	10	4	
0	1	5	1	
3	6	2	3	
1	3	10	0	

FFT Distorted Image				
56+0i	-21-7i	10+0i	-21+	
7+7i	-6-2i	7+5i	-12- 6i	
14+0i	7-1i	-12+0i	7+ 1i	
7-7i	-12+6i	7-5i	- 6+2	

Argument (Original Image)				
0	2.2035	0	- 2.2035	
-1.2490	1.2490	0.3805	2.6477	
3.1416	-2.6012	3.1416	2.6012	
1.2490	-2.6477	- 0.3805	-1.2490	

Argument (Enhanced Image)					
0	-2.8198	0	2.8198		
0.7854	-2.8198	0.6202	-2.6779		
0	-0.1419	3.1416	0.1419		
-0.7854	2.6779	-0.6202	2.8198		

Original – Enhanced				
0	-0.6163?	0	-0.6163	
0.4636	-1.5708	-0.2397	-0.0303	
3.1416	2.4593	0	2.4593	
0.4636	-0.0303	-0.2397	-1.5708	

 $SPE = \frac{1}{16} (0^{2} + 0.6163^{2} + 0^{2} + 0.6163^{2} + 0.4636^{2} + 1.5708^{2} + 0.2397^{2} + 0.0303^{2} + 3.1416^{2} + 2.4593^{2} + 0^{2} + 2.4593^{2} + 0.4636^{2} + 0.0303^{2} + 0.2397^{2} + 1.5708^{2}) =$ 

$$SPE = 28.20703848/16 = 1.7629 \tag{3.12}$$

The SPE result calculation shown in Figure 20 is equal to (3.14).

HOME PLOTS	APPS	EDITOR PUBLISH VIEW	🛃 🖥 🔏 🖄 🏦 🗇 😪 🗖 🕐 Search Docum	entation 🔎
ev Open Save Pint + FILE FLE		✓ Indent ✓ Inden		
urrent Folder		Z = DEMOP = ADDICOL ISIC =		G
Name +	Ŭ	SpectralPhaseError.m X +		
result8xt SignatoNoiseRatio.m SpectralMapatioAriseRatio.m SpectralPhaseEroor.m Structuralcontent.m Structuralcontent.m Structuralcontent.m Itest.ing.sciatd Ite	•	<pre>1 WFrogram for SpectralPhaseError Calculation 2 3 4</pre>		
orkspace		16 17 - error=abs(argffth)-abs(argfftB);		
ame AD Value AD 0.3750 afta 4x4 double aftb 4x4 double		<pre>SPE =sum(sum( error.*error)) / (M * N); </pre>		
ans 'distImg'		Command Window		
argGmagdist 4x4 double argGmagorg 4x4 double		New to MATLAB? See resources for <u>Getting Started</u> .		
disting 4x4 double distSiz [4,4]	<b>`</b>	3-spectral phase error = 1.7629 Æ		
			SpectralPhaseError	Ln 18 Col

Figure 20: MATLAB result of SPE calculation.

# 3.2.2.13 Implementation and Testing of SME (2.13)

Explanation of the SME code (each line number corresponds to its code in Appendix

A13):

Line 1: SME function has two inputs rel\_img which refers to the original image and enhcd\_img refers to the enhanced image, Line 4: we use a ready matlab function "**fft**" to transform the real image to the frequency domain, Line 5: shows the real part of the original image Line 6: shows the imaginary part of the original image Line 7: we take the SQRT of the real part summed with imaginary part of the original image, Line 8: shows the real part of the enhanced image, Line 9: shows the imaginary part of the enhanced image Line 10: we take the sqrt of the real part summed with imaginary part of the enhanced image Line 11: using the formula provided in (2.13) SME will be calculated.

Fourier Transform [13]:

Original Image				
0	1	7	3	
4	2	8	10	
0	2	1	6	
6	3	5	4	

FFT Original Image				
62 +0i	-11+15i	0+0i	-11-15i	
2-6i	1+3i	10+4	-13+7i	
-22+0i	-5-3i	-8+0i	-5+3i	
2+6i	-13-7i	10-4i	1-3i	

Distorted Image				
2	5	10	4	
0	1	5	1	
3	6	2	3	
1	3	10	0	

FFT Distorted Image				
56+0i	-21-7i	10+0i	-21+7i	
7+7i	-6-2i	7+5i	-12-6i	
14+0i	7-1i	-12+0i	7+1i	
7-7i	-12+6i	7-5i	-6+2i	

FFT (Original Image) Gradient:

Sqrt  $62^2 + 0^2 = 62$ .

Sqrt -11^2 + 15^2 = Sqrt 121+225 = Sqrt 346 = 18.6011.

Gradient (Original Image)				
62	18.6011	0	18.6011	
6.3246	3.1623	10.7703	14.764 8	
22	5.8310	8	5.8310	
6.3246	14.7648	10.7703	3.1623	

Gradient (Distorted Image)				
56	22.1359	10	22.1359	
9.8	6.3246	8.6	13.4164	
14	7.0711	12	7.0711	
9.8	13.4164	8.6	6.3246	

Original – Distorted				
6	-3.534	-10	-3.534	
-3.574	-3.162	2.168	1.348	
8	-1.24	-4	-1.24	
-3.574	1.348	2.168	-3.162	

 $SME = (6^{2} - 3.534^{2} - 10^{2} - 3.534^{2} - 3.574^{2} - 3.162^{2} + 2.168^{2} + 1.348^{2} + 8^{2} - 1.24^{2} - 4^{2} - 1.24^{2} - 3.574^{2} + 1.348^{2} + 2.168^{2} - 3.162^{2})/16$ SME = 295.795364/16 = 18.9(3.13)

The result of SME calculation shown in Figure 21 is equal to our manual calculation result (3.13).

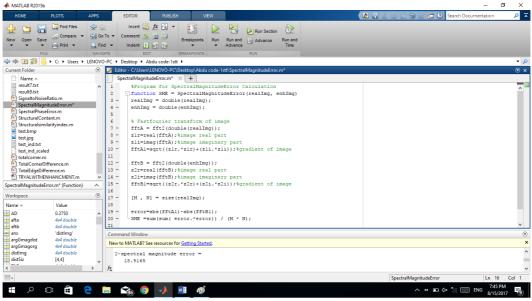


Figure 21: MATLAB result of SME calculation.

# 3.2.2.14 Implementation and Testing of Gradient Phase Error (GPE)

Explanation of the GPE code (each line number corresponds to its code in Appendix A-14):

Line 1: GPE function has two inputs rel\_img which refers to the real image and enhcd\_img refers to the enhanced image, Line 4: the gradient 2 of the original image will be calculated, Line 5: transfer gradient values of the original image to complex number, Line 6: angle of complex of the original image will be calculated, Line 7: the gradient of the enhanced image will be calculated, Line 8: transfer gradient values of the enhanced image to complex, Line 9: angle of complex of the enhanced image will be calculated Line 10: [m n] where m is amount of row the real image has and n is amount of columns the real image has; Line 11: difference between real image and enhanced image will be calculated and saved in 'eror' variable, Line 12: using the formula provided in (2.14) GPE will be calculated.

Gradient of The Original Image:

Fx1				
1	3.5	1	-4	
-2	2	4	2	
2	0.5	2	5	
-3	-0.5	0.5	-1	

Fy1			
4	1	1	7
0	0.5	-3	1.5
1	0.5	-1.5	-3
6	1	4	-2

Z =				
1+4i	3.5+1i	1+1i	-4+7i	
-2+0i	2+0.5i	4-3i	2+1.5i	
2+1i	0.5+0.5i	2-1.5i	5-3i	
-3+6i	-0.5+1i	0.5+4i	-1-2i	

$\mathbf{F} =$				
3-2i	4-4i	-0.5-5i	-6-3i	
1+0.5i	2.5+0.5i	0-4i	-4-0.5i	
3+0.5i	-0.5+1	- 1 5+2 5i	1-0.5i	
2-2i	4-3i	-1.5+8i	-10-3i	

Arg Original (Angle(Z))			
1.3258	0.2783	0.7854	2.0899
3.1416	0.2450	-0.6435	0.6435
0.4636	0.7854	-0.6435	-0.5404
2.0344	2.0344	1.4464	-2.0344

Arg Distorted (Angle(F))				
-0.5880	-0.7854	-1.6705	-2.6779	
0.4636	0.1974	-1.5708	-3.0172	
0.1651	2.0344	2.1112	-0.4636	
-0.7854	-0.5880	1.7561	-2.8501	

Ar	g Original -	- Arg Distor	·ted
0.7378	-0.5071	-0.8851	-0.5880
2.6779	0.0476	-0.9273	-2.3737
0.2985	-1.2490	-1.4677	0.0768
1.2490	1.4464	-0.3097	-0.8157

**GPE** = 1/16 (0.7378 ^2 -0.5071^2 -0.8851 ^2 -0.5880 ^2 + 2.6779 ^2 +0.0476 ^2 -0.9273 ^2 -2.3737 ^2 + 0.2985^2 - 1.2490 ^2 -1.4677 ^2 +0.0768 ^2 +1.2490 ^2 +1.4464 ^2 -0.3097 ^2 -0.8157^2) =

 $\mathbf{GPE} = (11.46483746 + 12.50639178)/16 = 1.48 \tag{3.14}$ 

The result of GPE calculation shown in figure 22 is equal to our manual calculation result (3.14).

📣 MATLAB R2015b		- o ×
HOME PLOTS APPS	EDITOR PUBLISH VEW	🔁 🖶 🔏 🖄 🖄 🗢 🗢 🔁 🕐 Search Documentation 🛛 🔎 🗖
New Open Save	30 To ▼ Comment % 22 A Breakpoints Run Run and Advance Run and Time	
	//GATE EDIT BREAKPOINTS RUN VO-PC ► Desktop ► Abdu code-1stt ►	ر ◄
Current Folder		© x
□ Name ♠	GradientohaseError.m X +	
Gradentplaseform (Function)     Gradentplaseform (Sinship)     Gradentplaseform     MaximumDifferencem     Ma	17 - argGmadist = angle(y);%angle of complex distorted image 18	
Name         Value           AD         0.3750           afta         4x4 double	20 21 - error=abs(argGmagorg)-abs(argGmadist); 22 - GFE =sum(sum(error.*error)) / (M * N);	
aftb 4x4 double	Command Window	6
argGmagdist 4x4 double	New to MATLAB? See resources for Getting Started.	
argGmagorg 4v4 double distIng 4v4 double distSiz [4,4]	5-gradient phase error = 1.4889 ft	
<b>-</b>		Ln 1 Col 1
📲 🔎 💷 🛍 🗎	🚍 💼 🌖 🧈 🗃 🛷	∧ м 🗈 ⊄× *⁄⁄₂ 📟 ENG 7:42 PM

Figure 22: MATLAB result of GPE calculation.

#### **3.2.2.15** Implementation and Testing of Gradient Magnitude Error (GME)

Explanation of the GME code (each line number corresponds to its code in Appendix A-15):

Line 1: GME function with two inputs rel\_img which refers to the real image and enhcd\_img refers to the enhanced image, line 4: using a ready MATLAB function the gradient (shows directional change in the intensity or color in an image) of the real image will be calculated, then the gradient magnitude is calulcated and saved in "gamgorg" variable, same steps will be applied for the enhanced image, Line 5: [m n] where m is amount of row the real image has and n is amount of columns the real image has; Line 6: difference between real image and enhanced image will be calculated and saved in "eror" variable, Line 7: using the formula provided in (2.15) GME will be calculated.

For testing our gme function used in our RFDS we will calculate manually the GME of one pixel only, which is the matrix element with value equal to (2).

0	riginal	Image (	I)
0	1	7	3
4	2	8	10
0	2	1	6
6	3	5	4

Gradient of Pixel (2,2)

$$Fx1 = (Df/Dx) = (8 - 4)/2$$
  
4 / 2 = 2

Sqrt  $2^2 + 0.5^2 =$  Sqrt 4.25 = 2.061

**Fy1** = (Df/Dx) = 
$$(2 - 1)/2$$
  
1 / 2 = 0.5

Fx1 3.5 1 -4 1 -2 2 2 4 2 0.5 2 5 -3 -0.5 0.5 -1

	F	y1	
4	1	1	7
0	0.5	-3	1.5
1	0.5	-1.5	-3
6	1	4	-2

	Orignal	Image I	
4.123	3.640	1.414	8.062
2	2.061	5	2.5
2.236	0.707	2.5	5.831
6.708	1.118	4.031	2.236

]	Distorted	Image '	Ι
3.605	5.656	5.024	6.708
1.118	2.549	4	4.031
3.041	1.118	2.915	1.118
2.828	5.408	8.139	10.440

	Original -	- Distorted	I
0.517	-2.016	-3	1.354
0.882	-0.488	1	-1.531
-0.805	-0.41	-0.415	4.712
3.879	-4.29	-4.1	-8.2

1/16 (0.517<sup>2</sup> + -2.016<sup>2</sup> + -3.610<sup>2</sup> + 1.354<sup>2</sup> + 0.882<sup>2</sup> + -0.488<sup>2</sup> + 1<sup>2</sup> + -1.531<sup>2</sup> + -0.805<sup>2</sup> + -0.41<sup>2</sup> + -0.415<sup>2</sup> + 4.712<sup>2</sup> + 3.879<sup>2</sup> + -4.29<sup>2</sup> + -4.1<sup>2</sup> + -8.2<sup>2</sup>)

GME = 1/16 (0.2673 - 4.0643 - 13.0321 + 1.8333 + 0.7779 - 0.2381 + 1 - 2.3440 - 0.6480 - 0.1681 - 0.1722 + 22.2029 + 15.0466 - 18.4041 - 16.8100 + -67.2400)GME = 164.2489/16 = 10.27(3.15)

The result of GME calculation shown in figure 23 is equal to our manual calculation result (3.15).

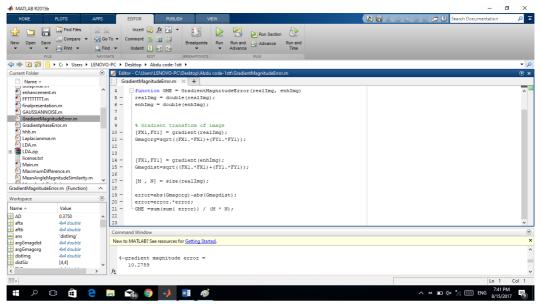


Figure 23: MATLAB result of GME calculation.

## 3.2.2.16 Implementation and Testing of Mean Angle Similarity (MAS)

Explanation of the MAS code (each line number corresponds to its code in Appendix B-16):

Line 1: MAS function with two inputs rel\_img which refers to the original image and enhcd\_img refers to the enhanced image, line 4: we use a ready MATLAB function "dot" to get the scalar dot of the real image and enhanced image, we also use another ready MATLAB function "acosd" which returns the inverse cosine of the element of the real; Line 5 : we take the sum of the variable "alfa2"; Line 6: using the formula provided in (2.16) MAS will be calculated.

	Origina	l Image	
0	1	7	3
4	2	8	10
0	2	1	6
6	3	5	4

	Distorte	d Image	
2	5	10	4
0	1	5	1
3	6	2	3
1	3	10	0

Alfa2 (Original Image, Distorted Image) = 0.9877, 0.9425, 0.6506, 0.9178

Angel2= Sum (Alfa2) = 3.4985

$$\mathbf{MAS} = 1 - (3.4985/16) = 0.7813 \tag{3.16}$$

The result of MAS calculation shown in Figure 24 is equal to (3.16).

📣 MATLAB R2015b						and a second sec	- 0 <mark>- ×</mark>
HOME PL	OTS APPS	EDITOR	PUBLISH VIEW			🖪 🔚 🔬 🗈 😒 🔄 🔁 🕐 Search Docume	ntation 🔎 🗖
New Open Save	Find Files Compare • Print • NAVIGATE		et te	Run and Advance RUN			
	C: ▶ Users ▶ user ▶ Desk		•				<del>م</del> +
Current Folder			litor - C:\Users\user\Desktop\iqm te				⊙ ×
Name  IMSENUMBERSTE			MASNUMBERTEST.m × LMSENU	JMBERSTEST.m 🛛 MAMSNUMBE	RTEST.m × +		
MAMSNUMBERTE  MASNUMBERTEST		1 2 3 4 - 5 - 6 7 8 - 9 - 10 - 11 12 - 13	realImg=[0,1,7,3; 4, enhImg=[2,5,10,4;0,1 Alfa2= 2/180 .* ( Angle2 = sum( Alf MAS = 1-Angle2/16			• norm(enhImg))));	
Details		^					
Workspace							
Name 🔺	Value						
🗄 Angle2	0.9877,0.9425,0.6506, 3.4985	-					
	1x4 double 1.3149	Comm	mand Window				Ŧ
	0.7813	E New 1	to MATLAB? See resources for Gettir	ng Started.			>
	4x4 double	>>	> MASNUMBERTEST				
	\$x4 double 10456		0.7813				
	10450 7952	- fx >>					

Figure 24: MATLAB result of MAS calculation.

# 3.2.2.17 Implementation and Testing of Mean Angle Similarity (MAMS)

Explanation of the MAMS code (each line number corresponds to its code in Appendix A-17):

Line 1: MAMS function with two inputs rel\_img which refers to the original image and enhcd\_img refers to the enhanced image, Line 4: we use a ready MATLAB function "dot" to get the scalar dot of the real image and enhanced image, we also use another ready MATLAB function "acosd" which returns the inverse cosine of the element of the real; Line 5 :"; Line 6 : using the formula provided in (2.17) MAMS will be calculated and assigned to variable "x"; Line 7: we take the sum of the variable "x; Line 8: we divide the sum and assign the result to variable "MAMS".

Alfa (Original Image, Distorted Image) = 0.9877, 0.9425, 0.6506, 0.9178

X= 1-(1-Alfa) \*(1-(Original Image-Distorted Image/255)) = 0.9883, 0.9452, 0.6667, 0.9215

SUM = Sum(X) = 3.5216

 $MAMS = (3.5216/16) = 0.2201 \tag{3.17}$ 

The result of MAMS calculation shown in Figure 25 is equal to (3.17).

MATLAB R2015b		- O - X
HOME PLOTS APPS	EDITOR PUBLISH VEW	Q
Image: Save of the save	o • Comment % ☆ ☆ ♪ Breakpoints Run Run and Advance Run and • Indent 5 ↔ Tor	
⊨ 파 🛅 🔀 🌗 🕨 C: 🕨 Users 🕨 user 🕨 De	iesktop ▶ iqm testing newww	-
Current Folder	⊙ 📝 Editor - C:\Users\user\Desktop\iqm testing newww\MAMSNUMBERTEST.m	$\odot$
🗋 Name 🔺	MASNUMBERTEST.m × LMSENUMBERSTEST.m × MAMSNUMBERTEST.m × +	
MULTERFUNDBERSTEST.m MAMASNUMBERTEST.m MASNUMBERTEST.m	<pre>1 % % function MAMS= MeanAngleMagnitudeSimilarity(realImg, enhImg) 2 3 - realImg=[0,1,7,3; 4,2,8,10:0,2,1,6;6,3,5,4]; 4 - enhImg=[2,5,10,4;0,1,5,1;3,6,2,31,3,10,0]; 5 6 - Alfa= 2/pi.* (acos(dot(realImg, enhImg)/(norm(realImg).* norm(enhImg)))); 7 - X = 1 - ((1-Alfa)* (1 - (norm(realImg-enhImg)/255)); 8 - Sum = sum(X); 9 - MAMS = Sum /16;</pre>	
	10 11 - disp(MAMS); 12	
Details	10 11 - disp(NAMS);	
Details Workspace	10 11 - disp(MGMS); 12	
Workspace Name  Value	10 11 - disp(MAMS); 12 •	
Workspace           Name ▲         Value           Alfa         [0.9877,0.9425,0.6506,           Alfa         [0.9877,0.9425,0.6506,	10 12 disp(MAMS);	
Workspace           Name         Value           Atfa         (0.9877,0.9425,0.6506,           Atfa         (0.9877,0.9425,0.6506,           Angle2         3.4985           enhimg         44/d couble	10 11 - disp (MAMS) ; 0 0 0 0 0 0 0 0 0 0 0 0 0	
Workspace           Name ▲         Value           Alfa         (0.9877, 0.9425, 0.6506,           Alfa2         (0.9877, 0.9425, 0.6506,           Angle2         3.4985           enhing         4xid double           LMSE         1.3149	10 12 disp(MAMS);	
Workspace           Alfa         (D.9877, 0.9425, 0.6506,)           Alfa         (D.9877, 0.9425, 0.6506,)           Angle2         3.4895           enhimg         4.64 double           MAKE         1.3149           MAMAN         0.2021	<pre>10 10 12 12 12 12 12 12 12 12 12 12 12 12 12</pre>	
Workspace           Name ▲         Value           Alfa         (0.9877, 0.9425, 0.6506,           Alfa2         (0.9877, 0.9425, 0.6506,           Angle2         3.4985           enhing         4xi double           LMSE         1.3149	10         11 -       disp (MGMS) ;         0         *         Command Window         New to MATLAS? See resources for <u>Getting Started.</u>	

Figure 25: MATLAB result of MAMS calculation.

# 3.2.2.18 Implementation and testing of Laplacian Mean Squared Error (LMSE)

Explanation of the LMSE code (each line number corresponds to its code in Appendix A-18):

Line 1: LMSE function with two inputs: rel\_img which refers to the real image, and enhcd\_img refers to the enhanced image;, line 4: [m n] where m is amount of row the real image has and n is amount of columns the real image has; Line 5& Line 6: we use a ready MATLAB function "del2" to calculate the second derivavtive of the real image and distorted image; Line 7: using the formula provided in (2.18) LMSE will be calculate.

0	Original Image										
0	1	7	3								
4	2	8	10								
0	2	1	6								
6	3	5	4								

Del	Del2 Original Image									
-6	2	-37	-49							
12	7	-12	-27							
-2	-2	17	17							
41	8	27	4							

C	Original Image										
0	1	7	3								
4	2	8	10								
0	2	1	6								
6	3	5	4								

Del2 Original Image									
-6	2	-37	-49						
12	7	-12	-27						
-2	-2	17	17						
41	8	27	4						

D	Distorted Image									
2	5	10	4							
0	1	5	1							
3	6	2	3							
1	3	10	0							

Del2 Distorted Image										
30	28	-18	-9							
19	12	-6	-14							
-24	-15	16	12							
12	-20	3	-54							

Del2 Or	Del2 Original Image - Del2 Distorted										
-36	-26	-19	-40								
-7	-5	-6	-13								
22	13	1	5								
29	28	24	58								

sum(sum(op-op2).^2)= 10456

sum(sum(op.^2))= 7952

LMSE = 10456/7952 = 1.3149.

(3.18)

The Result of LMSE calculation shown in Figure 26 is equal to (3.18).

A MATLAB R2015b	and a second	- 0 <mark>- ×</mark>
HOME PLOTS APPS	EDITOR PUBLISH VEW	ion 🔎
Image: Save     Image: Compare → Compar	Incert Comment % 22 The Presponse Run Run and Advance Run and The Comment Sector Run Run Run Run and The Run	
🛑 🔶 🛅 🔀 🌗 🕨 C: 🕨 Users 🕨 user 🕨 Deskto		-
Current Folder	Zeditor - CiUsers/user/Desktop/ligm testing newww/LMSENUMBERSTEST.m     MASNUMBERTEST.m × LMSENUMBERSTEST.m × + +	$\odot$
MLMSRUMBERTEST.m MAMSNUMBERTEST.m MASNUMBERTEST.m	<pre>i % function LMSE= LaplacianMeanSquaredError(realImg, enhImg)%lmse</pre>	
Details		
Workspace	•	
Name A Value		
enhlmg 4x4 double LMSE 1.3149 OP 4x4 double		
OP 4x4 double OP2 4x4 double	Command Window	
p 10456	New to MATLAB? See resources for <u>Getting Started</u> ,	
PX 7952 realImg 4x4 double	LMSENUMBERSTEST 1.3149	
	A »	

Figure 26: Matlab result of LMSE calculation.

For the rest of the IQMs (SSIM, VIF, RRED, HLFI, JQI, BIQI, AND NIQE) we implemented the following:

1. Downloaded ready implemented code from:

"live.ece.utexas.edu/research/quality/index\_algorithms.htm" Texas University [17].

We added the function files of every IQM to our thesis code folder. Then in our main.m file we call all the IQM functions. Input and output we get from these functions:

2. VIF

**Input:** (1) Img1: the reference image as a matrix. (2) Img2: the distorted image (Order Is Important).

**Output:** (1) VIF the visual information fidelity measure between the two images.

3. NIQE

Input: A test image loaded in an array.

**Output:** A quality score of the image. higher value represents a lower quality.

### 4. JPEG

Input: A test 8bits/pixel grayscale image loaded in a 2-d array.

Output: A quality score of the image. the score typically has a value. between 1 and

10 (10 represents the best quality, 1 the worst).

## 5. BIQI

Input: A test 8bits/pixel grayscale image loaded in A 2-D array.

**Output:** A quality score of the image. the score typically has a value. between 1 and 10 (10 represents the best quality, 1 the worst).

## 6. RRED

Input: A test 8bits/pixel grayscale image loaded in a 2-d array.

**Output:** Estimated entropy at different locations and the local spatial/temporal ('spatial' and 'temporal' refer to the spatial and temporal scaled entropy information for different blocks in the subband).

### 3.2.3 Implementation of Training Structure

MATLAB 2015 provides a classification learner application that can be used to import tables from the work space. The application is very simple to use. It automatically extracts predictors (our IQM's) and observations (real or fake image), also has more than one classification algorithm such as LDA and QDA which are used for classifying the extracted samples of an image.

In order to use classification learner application, the results have to be arranged in a table and displayed in the workspace. The results displayed in workspace will be all imported to the classification learner application, then we select the table which has all the fingerprint image results of IQM's, after that we select the response (Real/Fake) and predictors (Quality Measures). The steps of implementing our classifier using the classification learner application are shown in Figures 27-31.

HOME	PLOTS	APP	S E	EDITOR	PUBLISH	VEW						9000	Search Documentation	on 🖌
More In	stall Package App App	Curve Fitting	Optimization	PID Tuner	System Identification	Signal Analysis	Image Acquisition	Instrument S Control		AB Coder Applicati		ation Distribution	·	
	ILE						APP	s						
1	🔊 📙 🕨 D: 🕨	thesis2 🕨 Nev	v folder (2) 🕨	final code ⊁ i	magequalitymeasu	rewithpic 🕨 Ab	du code 🕨							
ent Fold	er		📝 Editor -	D:\thesis2\New	folder (2)\final cod	e\imagequalityn	neasurewithpic\A	bdu code\abdu4	.m					
ile		^	abdu.r	n ≍abdu4.	m × +									
nmand V	Vindow													
	5267 0285													
	MSE	SNR	PSNR	NK	AD	SC	MD	RAMD	NAE	TED	SME	SPE	GME	GPE
0.	00092134	29.666	78.487	0.99406	0.0009757	1.0109	0.24721	0.2189	0.014769	0.0055885	165.73	0.0023344	0.00084169	0.525
0	.0018056	25.909	75.565	0.98719	0.00097497	1.0236	0.30818	0.27437	0.030249	0.010141	336.39	0.001637	0.0017335	0.665
ο.	00092134	29.666	78.487	0.99406	0.0009757	1.0109	0.24721	0.2189	0.014769	0.0055885	165.73	0.0023344	0.00084169	0.525
0	.0018056	25.909	75.565	0.98719	0.00097497	1.0236	0.30818	0.27437	0.030249	0.010141	336.39	0.001637	0.0017335	0.665
0.	00077608	30.079	79.232	0.99434	0.00097595	1.0104	0.23563	0.22512	0.014612	0.0055781	137.54	0.0021246	0.00071072	0.41
0	.0010434	29.154	77.946	0.99401	0.00097608	1.0109	0.24728	0.22956	0.01595	0.0075729	189.12	0.002363	0.0010104	0.557
Ο.	00095809	28.114	78.317	0.99082	0.00097309	1.0171	0.24147	0.21345	0.024174	0.0056979	172.61	0.0021583	0.00085074	0.488
0.	00077608	30.079	79.232	0.99434	0.00097595	1.0104	0.23563	0.22512	0.014612	0.0055781	137.54	0.0021246	0.00071072	0.41
0	.0010909	28.682	77.753	0.99338	0.00097561	1.012	0.23743	0.21528	0.019604	0.0093906	198.53	0.002211	0.0010525	0.632
	.0011389	28.352	77.566	0.99255	0.00097496	1.0136	0.22575	0.21819	0.020312	0.0087188	207.1	0.0022274	0.0010683	0.545
	00061564	28.48	80.238	0.99175	0.0007184	1.0153	0.21268	0.18518	0.0269	0.0063333	111.15	0.0057324	0.00052643	0.383
	00065386	28.631	79.976	0.99049	0.00087616	1.018	0.20167	0.17304	0.026395	0.0064844	116.55	0.0036697	0.00057279	0.385
	00061564	28.48	80.238	0.99175	0.0007184	1.0153	0.21268	0.18518	0.0269	0.0063333	111.15	0.0057324	0.00052643	0.383
	00065386	28.631	79.976	0.99049	0.00087616	1.018	0.20167	0.17304	0.026395	0.0064844	116.55	0.0036697	0.00057279	0.385
	00083715	28.138	78.903	0.99067	0.00088737	1.0174	0.2066	0.18413	0.028246	0.0064427	151.76	0.0046852	0.00073305	0.491
	00086932	26.75	78.739	0.98739	0.00083905	1.0237	0.20167	0.18093	0.035568	0.0069479	158.29	0.0047787	0.00075073	0.375
		29.329	80.466	0.99208	0.00087389	1.0149	0.20167	0.17253	0.022648	0.0047917	102.95	0.0036381	0.00052062	0.446
ο.	00058409 00042173	29.188	81.881	0.9917	0.00079205	1.0156	0.20167	0.17447	0.025268	0.0046875	73.786	0.0084337	0.00036497	0.3652

Figure 27: Table which has all the fingerprint image results of IQMS we used.

Figure 27 shows the table created in the workspace with 100 fingerprint images, 50 fake and 50 real images. To create table to display our RFDS results we had to set a variable for every single IQM, each variable contains results calculated for one single IQM, then we used a ready MATLAB function called "table" for creating tables with our IQM variables in it. The table is imported to the classification learner application. Table is not completely shown in figure 27 due to screenshot limitation.

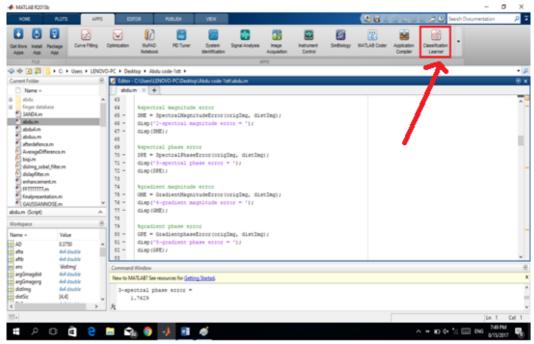


Figure 28: The arrow shows a button used to call the classification learner app in MATLAB.

Figure 28 shows all the apps provided in MATLAB 2015; to access the classification learner app in MATLAB click on apps, select classification learner app.

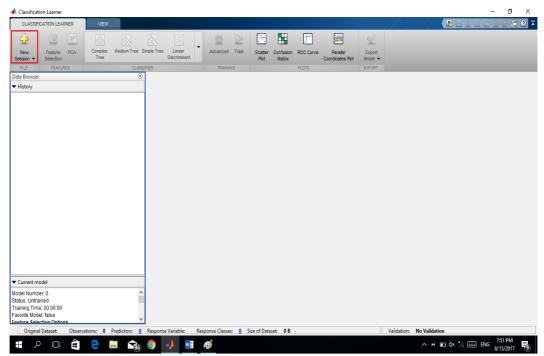


Figure 29: Start new session to import results.

Figure 29 shows classification learner first page, click new session to import all datasets from the recent work space.

Step 1	Step 2					Step 3						
Select dataset from MATLAB workspace.	Select predictors an	nd response.				Define validation method.						
i AD	Name	Туре	Range	Import as		Cross Validation						
RD GME	USERS	cell	2 unique	Response	~	0						
PE	MSE	double	0.000276086 0	Predictor	~	Protects against overfitting by partitioning the data set i						
aussF	SNR	double	24.0263 34.9059	Predictor	~	folds and estimating accuracy on each fold.						
	PSNR	double	75.7656 83.7204	Predictor	~							
19	NK	double	0.980054 0.997		v v							
D SE	AD	double	0.000811909 0		×	Cross-validation folds: 5 folds						
AE	sc	double	1.00396 1.03727									
π-	MD				$\sim$							
SNR		double	0.20167 0.302508		$\sim$							
	RAMD	double	0.151429 0.260		$\sim$							
AMD	NAE	double	0.00785197 0.0	Predictor	$\sim$							
C ME	TED	double	0.00480729 0.0	Predictor	~	Holdout Validation						
NB	SME	double	42.59 319.175	Predictor	~	Recommended for large data sets.						
PE	SPE	double	0.00148655 0.0	Predictor	×	recommended for all ge data colo.						
CD	GME	double	0.000284421 0	Predictor	~							
ED	GPE	double	0.276111 0.798		× ×							
isting	TCD	double	0.000606796 0		×	Percent held out: 25%						
stSiz	100	double	0.0000001000	Fiedicioi	~							
x						4 b						
-												
igImg												
igSiz												
						O No Validation						
						O No validation						
						No protection against overfitting.						
	¥											
Use columns as variables												
						Read about validation						
Use rows as variables												

Figure 30: Selection of our dataset, predictors and response in MATLAB.

Figure 30 shows the imported datasets from the works space which includes table t, the one we created, response (Real/Fake), and the predictors which is the image quality measures.

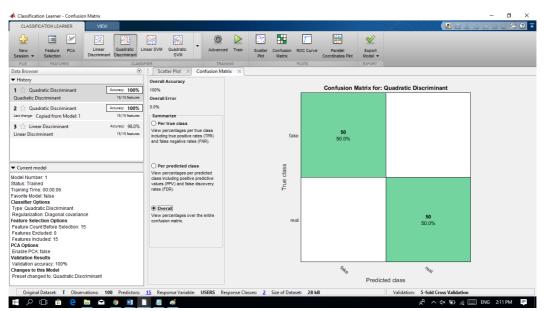


Figure 31: Selection of the classifier for our table imported and training result.

Figure 31 shows the classifiers that can be selected on the left-side of the screenshot, in our case we selected LDA classifier shown on the left corner of the figure, we then clicked train button to start the classification process for the fingerprint images which were selected for training. The right-side of Figure 29 shows the result of training 50 real and 50 fake fingerprint images, the training result had classified them all correctly. This training result obtained is considered as our training model, Figure 30 shows how to use and export the training model.

### **3.2.4 Implementation of The Classification Process**

The training model we got from training on 100 fingerprint images is required for classification process. We input 1000 images from the fingerprint database for RFDS to classify the images on real and fake. There are four stages to implement classification process.

Export model of the classifier to the workspace to classify fingerprint images. Figure
 30 shows how to use and export the training model

2. To classify the fingerprint images using the training model obtained from the classification learner app, we need to call a ready MATLAB function "yfit = trainedclassifier.predictfcn(t)" and insert it after the table creation code, because this function will predict the fingerprint images if they are real or fake using a table with all the IQM results listed in it, in our case the function will predict table 'T' results real or fake based on the training model we obtained.

3. 1000 fingerprint images were used for classification. (Appendix C2) shows results of the 1000 images classification.

4. Run and classify.

The steps that show how the classification process works are shown in Figures 32-35: Click on the rectangle highlighted on the top right corner of Figure 32 to export the training model of 100 real and fake fingerprint training images, we export this model to our work space for the classification process. We enter then random images of fingerprints from different users and let our training model classify if the images FED to the RFDS are real or fake. For exporting the model click on the top right button "export model" and select the first option export model", by this, the training model will be exported to the workspace.

CLASSIFICATION LEARNER VIEW						1999			<i>A</i> ##\$\$\$\$\$		1 i 9 c i i
🕂 🗉 🗹 🐹			٢			<u>/</u>		<b>V</b>			
New Feature PCA Linear ession - Selection Discriminant	Quadratic Line Discriminant	ear SVM Quadratic SVM	Advanced	Train Scatt Plot		ROC Curve	Parallel Coordinates Plot	Export Model 👻			
FILE FEATURES	CLASSIFI		TRAIN			PLOTS		Expo	rt Model		
ta Browser		Scatter Plot 🛛 🕹	Confusion Mat	rix X ROC Cu	rve 🛛 Pa	rallel Coordina	ites Plot 🛛 🕹		rt the currently selected cla		
History		Variable on X axis:							ry list to the workspace to	make predictions wi	th new data
🟠 Linear Discriminant 🛛 🖌	ccuracy: 96.7%	MSE	~				Scatter Pl	Expo	rt Compact Model		
near Discriminant	15/15 features		_						rt the currently selected cla		
C Quadratic Discriminant	couracy: 100%	Variable on Y axis:						with	out its training data to the	workspace to make	predictions with new da
Juadratic Discriminant	15/15 features	SNR	~	34	•			Gene	erate Code		
		JAR	•		•				rate MATLAB code for train		
	ccuracy: 100%	Legend				1		class	ifier in the History list, inclu	ding validation pred	dictions
st change: Copied from Model: 2	15/15 features	_		32		•					
		Correctly classified fake		32	-	•					
		• real									
						1 <b>.</b>					
Current model		Misclassified - true c	ass is:	~ 30		4					
del Number: 2		× real		SNR 30		••	• • •				
tus: Trained				0)			•	•			
ining Time: 00:00:00 orite Model: false		Show Classifier Res	ults								
ssifier Options		Classifier Results		28	-						
pe: Quadratic Discriminant		Color of misclassified	points				•				
gularization: Diagonal covariance Iture Selection Options		represents:							• •		
ature Count Before Selection: 15		True class	~	26						••••	
atures Excluded: 0		1100 0033								•	•
atures Included: 15										• •	•
A Options able PCA: false											
idation Results				24	-					•	
lidation accuracy: 100%						0.4	0.6 0	1.8	1 12	1.4	1.6
anges to this Model eset changed to: Quadratic Discriminant									MSE		× 10 <sup>-3</sup>
eset changed to, quadratic Discriminant											~ 10

Figure 32: Exporting training model of 100 real and fake fingerprint images.

HOME PLOTS APPS	EDITO	DR PUE	BLISH VIEW						862	🖻 🔁 🕐 S	earch Documentation	1 <mark>/</mark>
t More Install Package App App	Optimization	MuPAD Notebook	PID Tuner System Identification	Signal Analysis	Image Acquisition	Instrument Control	SimBiology	MATLAB Coder	Application Compiler	Classification Learner	•	
FILE				AF	PPS							
Image: Section 1.												•
rrent Folder 🐨	Editor - C		-PC\Desktop\Abdu code-1		ancement.m	E FETTTTTT	m × hhh.r	]	]			۲
Name A abdu finger database abdu abdu abduAm abduAm abduAm abduAm bduAm bduAm bduAm bduAm abduAm bduAm bduA	283 - 284 - 285 - 286 - 287 - 289 - 289 - 290 - 291 - 292 - 292 - 293 - 294 - 295 - 295 - 296 -	<pre>PSNR=[w(3,1) PSNR=[w(4,1);: AD=[w(5,1);: SC=[w(6,1);: MD=[w(7,1);: RAMD=[w(8,1) TED=[w(10,1] SPE=[w(11,1] SME=[w(12,1] GME=[w(13,1] GPE=[w(14,1] TCD=[w(15,1] T=table(USE]</pre>	N         Boldudm         X all X           V(x)(x)(x)(x)(x)(x)(x)(x)(x)(x)(x)(x)(x)(	<pre>3,2):w(3,5):w(3,5):w(2);w(5,5):w(2);w(5,5):w(2);w(7,5):w(3,2):w(7,5):w(3,2):w(3,5):w(4,1);2):w(10,2):w(10,2):w(10,2):w(11</pre>	<pre>w(3,6);w(3 4,6);w(4,7) 5,6);w(5,7) 5,6);w(5,7) 7,6);w(7,7) w(8,6);w(8, (9,6);w(9, ,5);w(10,6) ,5);w(10,6) ,5);w(11,6) ,5);w(12,6) ,5);w(13,6) ,5);w(14,6)</pre>	7);w(3,8) ;w(4,8);w ;w(5,8);w ;w(5,8);w ;w(5,8);w ;w(7,8);w 7);w(9,8); ;w(10,7); ;w(11,7); ;w(12,7); ;w(12,7); ;w(14,7); ;w(15,7);	;w(3,9);w( (4,9);w(4, (5,9);w(5, (7,9);w(5, (7,9);w(7, ;w(8,9);w( w(10,8);w( w(11,8);w( w(11,8);w( w(11,8);w( w(12,8);w( w(14,8);w( w(15,8);w(	3,10);w(3,1) 10);w(4,11) 10);w(5,11) 10);w(5,11) 10);w(7,11) 8,10);w(7,11) 8,10);w(7,11) 10,9);w(10, 11,9);w(11, 12,9);w(12, 13,9);w(12, 14,9);w(14,	;w(4,12); ;w(5,12); ;w(5,12); ;w(6,12); ;w(7,12); 1);w(8,12); 10);w(9,12); 10);w(11,1 10);w(11,1 10);w(12,1 10);w(13,1 10);w(13,1 10);w(15,1)	<pre>v(4,13):w(4, v(5,13):w(5, v(6,13):w(5, v(6,13):w(7, ):w(8,13):w(7, ):w(8,13):w(7, 11):w(10,12; 11):w(10,12; 11):w(11,12; 11):w(12,12; 11):w(12,12; 11):w(14,12; 11):w(14,12; 11):w(15,12;11):w(15,12; 11):w(15,12;11):w(15,12; 11):w(15,12;11):w(15,12; 11):w(15,12;11):w(15,12; 11):w(15,12;11):w(15,12; 11):w(15,12;11):w(15,12;11):w(15,12; 11):w(15,12;11):w(15,</pre>	<pre>14);w(4,15);w ,14);w(5,15);w ,14);w(6,15);w ,14);w(7,15);w (8,14);w(9,15); );w(10,13);w(1 );w(11,13);w(1 );w(12,13);w(1 );w(12,13);w(1 );w(14,13);w(1</pre>	(4,16);w (5,16);w (6,16);w (7,16);w ;w(8,16); w(9,16); 0,14);w( 1,14);w( 2,14);w( 3,14);w( 4,14);w(
orkspace (*		disp(T);										
ame 🔺 Value	299											
AD 60x1 double ans 'diating' bmpfiles 50x1 struct disting 480x60 double distSis [480,400] Gauss 480x60 double GME 60x1 double GME 60x1 double CPE 60x1 double	Command W New to MAT Variab: To use	LAB? See resource les have been the exported	es for <u>Getting Started</u> . en created in the ed classifier 'tra lassifier.predictF	inedClassifi		• predicti	ons on new	data, T, u	se:			,

Figure 33: After exporting the training model, we can use it to make prediction by the code provided in the command window highlighted with a rectangle.

Figure 33, shows that our training model has been exported to the works space, as it can be seen clearly in the command window. If this highlighted rectangle message

doesn't appear in the command window, it means, the training model wasn't exported successfully.

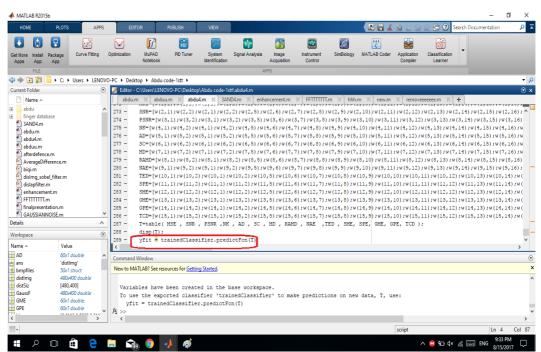


Figure 34: The highlighted rectangle shows the code used to classify images according to our exported training model.

Figure 34, Shows the code used for classifying the images "yfit = trainedclassifier.predictfcn(t)". As we mentioned before this code has to be added after creating a table with all the results of the IQMS listed in it, so that we can classify the fingerprint images as real or fake. In our case we created a table named 'T' with all the results saved, then we added the function "yfit = trainedclassifier.predictfcn(t)" in our code to classify images results from the table 'T' according to our exported training model.

📣 MATLAB R2015b													-	ð ×
HOME	PLOTS APP	PS EDI	TOR	PUBLISH	VEW						<b>b i</b> 5	@ 🖯 🕐 s	earch Documentatio	n 🔎 🔺
Get More Install Pa	Curve Fitting	Optimization	MuPAD Notebook	PID Tuner	System Identification	Signal Analysis	Image Acquisition	Instrument Control	SimBiology	MATLAB Coder	Application Compiler	Classification Learner	•	
FILE						A	PPS							
💠 🄶 🖪 🔁 📒	C: → Users → LEN		- · · · · · · · · · · · · · · · · · · ·											م -
Current Folder		0				>\Abdu code\abd	u4.m							🖲 🗙
🗋 Name 🔺			- /	ı.m ≍∫abdu4	m × [ + [									
			Comman	d Window										0
3AND4.m	50		New to N	/ATLAB? See reso	urces for <u>Getti</u>	ng Started.								×
abdu.asv			ALTO	-										^
abdu.m abdu4.asv				'real'										
abdu4.asv				'real'										
abduu.m				'real'										
afterdefence.				'real' 'real'										
AverageDiffer	ence.m			'real'										
dislmg_sobel	filter m			'real'										
🖄 dislapfilter.m				'real'										
enhancement	t.m		<b>.</b>	'real'										
PFTTTTTT.m Details		~	_	'real'										
			_	'fake' 'fake'										
Workspace		6	0	'fake'										
Name 🔺	Value			'fake'										
AD AD	20x1 double		^	'fake'										
🔤 ans 📑 bmpfiles	'distImg' 272x1 struct			'fake'										
distImg	480x400 double			'fake' 'fake'										
distSiz	[480,400]			'fake'										
GaussF	480x400 double			'fake'										
GME GPE	20x1 double 20x1 double													
H .	[0.0113,0.0838,0.011	3:	<i>f</i> <b>x</b> >>											~
	10		< <											>
•														

Figure 35: Results of 10 real and fake images classified correctly using the classification code.

Figure 35 shows the results obtained from fingerprint images after running the classification code (Appendix A). In this run we had 10 real images and 10 fake all were classified correctly. The fingerprint images used were obtained from fingerprint database atvs-ffp db [33]. Figure 36 shows a screenshot of one real and fake fingerprint image sample we used in our experiments. Real and fake finger print images were read from directory" c:\users\lenovo-pc\desktop\hp\_new folder (2)\abdu code\finger database\abdu\lbiometrika\tm real". Appendix A. Line 3. Shows the code we used for reading the directory of the fingerprint images.

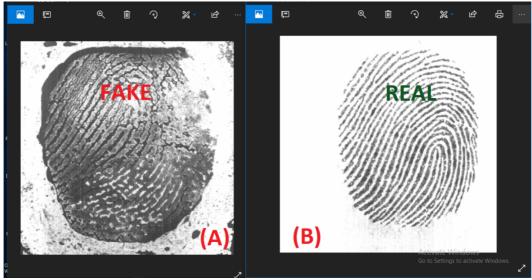


Figure 36: Fingerprint images used in our expirments, (A) fake fingerprint image from the file" U01\_F\_Fc\_Li\_01.Bmp", (B) real fingerprint image form the file" U01\_O\_Fc\_Li\_01.Bmp".

# 3.3 Development of A Method of Defining the Best IQMs

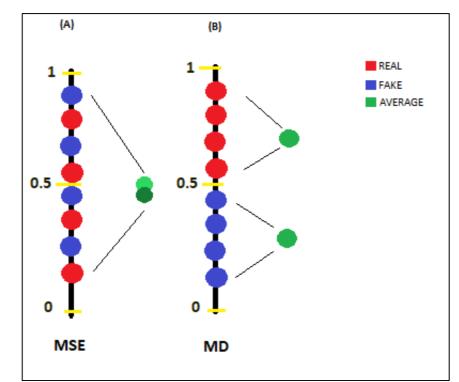


Figure 37: Numerical axis for MSE (A) And MD (B), showing MSE of fake (real) fingerprint images. And the respective averages calculated by (3.16) and (3.17).

Figure 37, shows a drawing that will help you to understand our basic idea behind finding the best 5 and 10 IQMs, as you can see in Figure 34 there are two graphs (A) for MSE and (B) for MD, both of the graphs have red dots which represent the values of MSE and MD in real fingerprint images, the blue dots which represent the fake fingerprint image values of MSE and MD, and the green dots represent the average of MSE and MD in real fingerprint images (red dots) and fake fingerprint images (blue dots) calculated by (3.16) and (3.17). Figure 34 has two scenarios: let's start with (A), the MSE red points are close to the blue points which means that the real and fake points in MSE are very close to each other and makes it difficult for us to distinguish between both of the points, in (A) the average point of the real points of MSE and the average point of the fake points of MSE are very close, in other words, it means that the real and fake points are very near to each other and it's hard to differentiate between them. In (B) the MD red and blue points are well separated from each other, so that means the real points and fake points can be well distinguished from each other, we can also see that in (B) the average points for the real and fake images are far away from each other, so (B) is far better than (A) in distinguishing real and fake fingerprint images, from these two cases we can say that MD is a better IQM than MSE. The IQM which will have the highest absolute value of the difference between the green dots representing averages will be set as the best in the list of the IQM sorted by the difference (3.18) in the descending order.

1. To find the best 5 and 10 IQM of our RFDS we need to apply the following steps for every measurement.

Find the IQM average of real fingerprint images (
$$\overline{\mathbf{x}}\mathbf{r}$$
).  
 $\bar{\mathbf{x}}\mathbf{r} = \frac{\sum_{i=1}^{N} x_i^r}{Nr}$ 
(3.16)

Where  $x_i^r$  is the value of IQM in one single real fingerprint image, NR is number of real fingerprint images FED to the system.

2. Find the IQM average of fake fingerprint images ( $\overline{x}f$ ).

$$\bar{x}f = \frac{\sum_{j=1}^{N} x_j^f}{Nf} \tag{3.17}$$

Where  $x_i^r$  is the value of iqm in one single fake fingerprint image, nf is number of fake fingerprint images fed to the system.

3. Find the difference between real  $(\bar{x}r)$  and fake fingerprint images  $(\bar{x}f)$ .

$$\Delta X = |\bar{x}r - \bar{x}f| \tag{3.18}$$

Apply these three steps for every single IQM, then sort them from maximum to minimum difference value which is found in step 3. The best IQM will be the one with the maximum distance value.  $\bar{x}r$ .

**Sort** = 
$$max(\Delta Xr \dots \Delta Xn)$$

Where n is the number of iqm used in the system

### 3.3.1 Implementation and Testing of The Method of Defining the Best IQMs

The code for the defining the best IQMs was implemented using MATLAB. The complete code is shown in Appendix B.

Explanation of defining the best IQMS code (Appendix B-1):

Line 1: We set the results of the IQM from 1-500 as real values. Line 2: we set the results of the IQM from 501-1000 as fake values. Line 3: we take the average of the real values of the IQM. Lines 4 and 5: display the average of the real values of the IQMs. Line 6: we take the average of the fake values of the IQM. Lines 7 and 8: display the average of the fake values of the IQM; line 9: displays the difference of the

averages of real values and fake values of the IQMs. Line 10: we take the absolute value of the difference calculated. Line 11: we display the result.

Explanation of ranking the best IQMs code (Appendix B-2):

Line 1: We list the names of all IQMs in one array. Line 2: we list the difference of the averages of real and fake values of every IQM in one array. Line 3: we create a table with "title" and "best 15" as its fields. Line 4: we sort the table rows in descending order.

### **3.3.1.1 Calculation of Defining the Best 5 IQMs**

### 1. For MSE

Real fingerprint image values of the MSE: 0.0017764, 0.0016535, 0.0016386, 0.00163, 0.0016254

Fake fingerprint image values of the MSE: 0.0015887, 0.0015395, 0.0015314, 0.0014846, 0.0014052

Average of the real fingerprint image values of MSE: (0.0017764 + 0.0016535 + 0.0016386 + 0.00163 + 0.0016254)/5 = 0.0017

Average of the fake fingerprint image values of MSE: (0.0015887 + 0.0015395 + 0.0015314 + 0.0014846 + 0.0014052)/ 5 = 0.0015

Difference between the average of real fingerprint image values of MSE and average of the fake fingerprint image values of MSE: |0.0017 - 0.0015| = 0.0001549

### 2. For SNR

Real fingerprint image values of the SNR: 32.608, 31.179, 30.719, 30.149, 30.149 Fake fingerprint image values of the SNR: 29.958, 29.674, 29.664, 29.584, 29.674 Average of the real fingerprint image values of SNR: (32.608 + 31.179 + 30.719 + 30.149 + 30.149) / 5 = 30.9608 Average of the fake fingerprint image values of SNR: (29.958 + 29.674 + 29.664 + 29.584 + 29.674) / 5 = 29.7108

Difference between the average of real fingerprint image values of SNR and average of the fake fingerprint image values of SNR: |30.9608 - 29.7108| = 1.2500

### 3. For PSNR

Real fingerprint image values of the PSNR: 81.171, 81.106, 80.41, 79.938, 79.541 Fake fingerprint image values of the PSNR: 79.142, 79.142, 78.689, 78.668, 78.522 Average of the real fingerprint image values of PSNR: (81.171 + 81.106 + 80.41 + 79.938 + 79.541) / 5 = 80.4332

Average of the fake fingerprint image values of PSNR: (79.142 + 79.142 + 78.689 + 78.668 + 78.522) / 5 = 78.8326

Difference between the average of real fingerprint image values of psnr and average of the fake fingerprint image values of PSNR: |80.4332-78.8326| =1.6006

#### 4. For NK

Real fingerprint image values of the NK: 0.99731, 0.99604, 0.99542, 0.99542, 0.9948 Fake fingerprint image values of the NK: 0.99453, 0.99447, 0.99417, 0.99409, 0.994 Average of the real fingerprint image values of NK: (0.99731 + 0.99604 + 0.99542 + 0.99542 + 0.9948) / 5 = 0.9958

Average of the fake values of NK: (0.99453 + 0.99447 + 0.99417 + 0.99409 + 0.994) / 5 = 0.9943

Difference between the average of real fingerprint image values of NK and average of the fake fingerprint image values of NK: |0.9958 - 0.9943| = 0.0015

### 5. For AD

Real fingerprint image values of the AD: 0.00097608, 0.00097608, 0.00097608, 0.00097608, 0.00097608, 0.197608

Fake fingerprint image values of the AD: 0.297608, 0.07608, 0.208, 0.1608, 0.397608 Average of the real fingerprint image values of AD: (0.00097608 + 0.00097608 + 0.00097608 + 0.197608) / 5 = 0.0403

Average of the fake fingerprint image values of AD: (0.297608 + 0.07608 + 0.208 + 0.208)

0.1608 + 0.397608) / 5 = 0.2280

Difference between the average of real fingerprint image values of ad and average of the fake fingerprint image values of AD: |0.0403 - 0.2280| = 0.1877

The best IQM is the one with the fake fingerprint image values. The rank of the best 5 IQMs will be:

- SNR 1.6006
- SNR 1.25
- AD 0.18772
- NK 0.001546
- MSE 0.0001549

A MATLAB R2015b		– o ×
HOME PLOTS APPS EDITO	PUBLISH VEW 🔃 🔂 🖬 🖉	🕼 🛱 😏 🖻 🔁 🕐 Search Documentation 🛛 🔎 🗖
New Open Save	at C for Far → ES → Paralports Run Recton D → Paralports Run Run and Advance Run and Time	ام -
	Z Editor - XXXXXXX.m 💿 🗙 Z Variables - tblA	
Name A B Constraints B Constra	Est_10m         X         bet25reakhetx22m         X         bbbjhdelm         X         United         X         bet35_destification.m         X         XXXX           90         -         F_AD         = AD(6:10);	000m × bestforrkarv × TH4_TEST_15m × +
Workspace 💿	Command Window	•
Name A         Value           AD         (9):708e-04;3:708e-0         A           Avg. F.AD         0.2280         A           Avg. F.MSE         0.015         A           Avg. F.MSE         0.015         A           Avg. F.NK         0.8326         A           Avg. F.NK         0.9433         A           Avg. F.NK         23:708         A           Avg. F.NK         0.0403         A           Avg. F.NK         0.0955         A	tile         best_IQM           'PSNR.'         1.6006           'NR.'         1.25           'AD.'         0.18772           'NK.'         0.001546           'MSE.'         0.0001545	^
Ave P DSNP 20.4222	n >>	script Ln 107 Col 3

Figure 38: The list of the best IQMs (in the red rectangle).

Figure 38 shows a screenshot with the list of the best IQMs. Using MATLAB, we inserted the same values used for every IQM in the manual calculation above and applied the same steps of calculation, which generated expected results at the end. For our RFDS expirment we used the same method explained above to define our best 15, 10, and 5 IQMs. Figure 38 shows a screenshot of our best 15, 10, and 5 IQMs generated by our code which is available in Appendix B. The MATLAB screenshot in Figure 39 shows our IQMs listed as follows: for **best 10 IQMs**: SME, PSNR, SNR, GPE, RAMD, MD, TCD, NAE, SPE, and SC, and the first five of them are the **Best 5 IQMs**.

📣 MATLAB R20156		– ø ×
HOME PLOTS APPS	EDITOR PUBLISH VIEW	🛃 🔚 🔏 🕸 🛍 🗇 🗭 🔁 🕐 Search Documentation 🛛 🔎 🗖
Image: Prind Files     Image: Prind Files       New     Open       Save     Image: Prind Files       Image: Prind Files     Image: Prind Files       Image: Prind Files     Image: Prind Files       Image: Prind Files     NAVIGATE	Breakpoints Run Run and S Advance Run and	
💠 🛅 🔀 📙 🕨 C: 🕨 Users 🕨 LENOVO-PC	Desktop > hp_New folder (2) > Abdu code >	ر 🕶
Current Folder	Editor - TM_Create_15.m	🕤 🗙 🌌 Variables - yfit
🗋 Name 🔺	TM_Create_15.m × TEST_15_(1000).m × +	
Structuralizationshipmides, m     Structuralizationshipmides, m     testipg     testipg     testing, solution     testind, solution     ThM_SectS-m     TM_Rest_S-m     T	<pre>     Command Window     tblB =         title</pre>	, (
TEST_15_(1000)un (Script)           Workspace           Wankspace           Trained Lassifier           Trained Lassifier           Trained Cassifier           Trained Cassifier           Trained Cassifier           Trained Cassifier           Trained Cassifier           Trained Cassifier           Total Cassifier           Trained Cassifier		
w 15:500 double x 500 yft 1000x1 cell z 500	*KSE.' 0.00030369 'AD.' 0.00011988 ✓ <	Activate Windows Go to Settings to activate Windows.

Figure 39: List of our best 15,10,5 IQMs used in our RFDS expirments.

# 3.4 Summary

In this chapter we have presented the implementation of our RFGS, including Gaussian Filter, Feature Extraction, Measurnment of Traning Strcture, Classification Process and the best IQMs using MATLAB r2015b as showen in (Appendices A- C).

# **Chapter 4**

# **EXPERIMENTS ON RFDS**

### **4.1 Experimental Setup**

The experiment was implemented on a PC with Windows 10 pro, 64-bit operating system, CPU Intel Core I7 2.50 Ghz, 8 Gb Ram, and MATLAB r2015b.

### 4.2 Database Used

ATVS fingerprint database [33] contains two datasets of fingerprint images with bmp file format:

**1. Dataset 1:** fake samples were captured from gummy fingers generated with the cooperation of the user. It contains fingerprint samples of the middle and index finger of both hands of 17 users, which means in total 68 (17x4=68) different fingers. Four samples of each fingerprint (Fake and Real) were captured in one acquisition session with the use of three sensors, so the dataset consists of 68 x 4 x 3 =816 real image samples and as many fake images.

The three types of sensors are used in the dataset:

- Biometrika Sensor.
- Thermal Sensor.
- Flat Capacitive Sensor.

**2. Dataset 2:** ds\_withoutcooperation fake samples were captured from gummy fingers generated with the cooperation of the user. It contains fingerprint samples of the

middle and index fingers of both hands of 16 users, which means in total 64 (16x4) different fingers. Four samples of each fingerprint (Fake and Real) were captured in one acquisition session with the use of three sensors, this way the dataset comprises 768 (64 x 4 x 3 = 816) real image samples and as many fake images.

In all the experiments we ran on the RFDS we used 50 real and 50 fake images for training stage, while in the classification stage we used 500 original fingerprint images and 500 fake fingerprint images a total of 1000 fingerprint images. Three different types of experiments on the RFDS were conducted: RFDS training and classification with 25 IQMs, RFDS training and classification with best 15 IQMs, RFDS training and classification with the best 10 IQMs, RFDS training and classification with the best 5 IQMs.

### **4.3 Classification Methods Used**

- LDA (Linear Discriminant Analysis)
- QDA (Quadratic Discriminant Analysis)
- Linear SVM, Quadratic SVM
- Logistic Regression, and fine KNN.

### 4.4 Code Parts for Experiments Conducting

The experiment code is divided into three parts:

- Input the Fingerprint Image and Filtering Code.
- Feature Extraction Code.
- Classification Process.

In the first part of the code (Appendix A, lines 1-29), we input the fingerprint images and convert them from RGB to grayscale image. Then, we convert the image to double, and apply a 3\*3 Gaussian filter to it; we resize both the original and fake fingerprint image to an equal size.

In the second part of the code (Appendix A, Lines 30-80), 25 image quality measures are implemented and each of them has a function for calculation, using the two inputs, real and enhanced fingerprint image. Each image quality measure function provided in the code was successfully tested. For training, 50 real fingerprint images and 50 fake fingerprint images with their 25 image quality measures calculated will be gathered all in a vector and arranged in a table to be able to import them to the classification learner application for training process.

In the third part which is the classification process, we use a user-friendly MATLAB application called "Classification Learner Application" to classify fingerprint images to real and fake images using six classifiers.

# 4.5 Experimental Results

## 4.5.1 Experiments on the RFDS Training and Classification with 15 IQMs

<b>15 IQMs</b> SME, PSNR, SNR, GPE, RAMD, MD, TCD, NAE, SPE, SC, TED, NK, MSE, GME, AND AD; ATV Database, 50 Real And 50 Fake Fingerprints.						
Classifier/Sample No.	FFR	FGR	HTER			
Lda/1000 Images	4%	30%	17%			
Qda/100 Images	4%	24%	14%			
Linear Svm/100 Images	2%	2%	2%			
Quadratic Svm/100 Images	0%	0%	0%			
Logistic Regression/100 Images	0%	0%	0%			
Fine Knn/100 Images	0%	0%	0%			

Table 4: Training results with 15 IQMs.

The results listed in Table 4, are generated from ATV database. There were 50 real and 50 fake fingerprint images input to the system to train, the detailed results of the 100 samples of fingerprints using 15 IQMs and 6 classifiers are shown in Appendix C-1, Figures 38-43.

<b>15 IQMs</b> SME, PSNR, SNR, GPE, RAMD, MD, TCD, NAE, SPE, SC, TED, NK, MSE, GME, AND AD; ATV Database, 500 Real And 500 Fake Fingerprints							
Classifier/Sample No. FFR FGR HTER							
LDA/1000 Images	5.4%	21.6%	13.5%				
QDA/1000 Images	12.4%	17.2%	14.8%				
Linear SVM/1000 Images	8.4%	7.6%	8%				
Quadratic SVM/1000 Images	1.6%	10%	5.8%				
Logistic Regression/1000 Images	8%	9%	8.5%				
Fine KNN/1000 Images	2.6%	13.6%	8.1%				

Table 5: Classification results with 15 IQMs.

Table 5, shows the classification results with 15 IQMs for 500 real and 500 fake fingerprint images inputted to the system to classify, the detailed results of the 1000 samples of fingerprints for classification are shown in Appendix C-2, Figures 44–49.

## 4.5.2 Experiments on The Rfds Training and Classification with Best 10 IQMs

<b>Best 10 IQMs</b> SME, PSNR, SNR, GPE, RAMD, MD, TCD, NAE, SPE, AND SC						
Classifier/Sample No.	FFR	FGR	HTER			
LDA/100 Images	4%	24%	14%			
QDA/100 Images	4%	12%	8%			
Linear SVM/100 Images	2%	0%	1%			
Quadratic SVM/100 Images	0%	0%	0%			
Logistic Regression/100 Images	0%	0%	0%			
Fine KNN/100 Images	0%	0%	0%			

Table 6: Training results with best 10 IQMs.

The results listed in Table 6, are generated with the use of 10 IQMs, there were 50 real and 50 fake fingerprint images inputted to the system to train, the detailed results of the 100 samples of fingerprints using 6 classifiers are shown in Appendix C-3, Figures 50-55.

Table 7: Classification results with best 10 IQMs.

<b>Best 10 IQMs</b> : SME, PSNR, SNR, GPE, RAMD, MD, TCD, NAE, SPE, AND SC							
Classifier/Sample No.	FFR	FGR	HTER				
LDA/1000 Images	9%	18.8%	13.9%				
QDA/1000 Images	13.4%	13.6%	13.5%				
Linear SVM/1000 Images	8.8%	7.4%	8.1%				
Quadratic SVM/1000 Images	3%	9.4%	6.2%				
Logistic Regression/1000 Images	14.6%	12.2%	13.4%				
Fine KNN/1000 Images	3.8%	13.2%	8.5%				

Table 7, shows the classification results with 10 IQMs for 500 real and 500 fake fingerprint images inputted to the system to classify. The results of the 1000 samples of fingerprint classification are shown in Appendix C-4, Figures 56–61.

### 4.5.3 Experiments on The RFDS Training and Classification with Best 5 IQMs

<b>Best 5 IQMs:</b> SME, PSNR, SNR, GPE, AND RAMD							
Classifier/Sample No.	FFR	FGR	HTER				
LDA/100 Images	4%	16%	10%				
QDA/100 Images	4%	6%	5%				
Linear SVM/100 Images	2%	0%	1%				
Quadratic SVM/100 Images	2%	0%	1%				
Logistic Regression/100 Images	0%	0%	0%				
Fine KNN/100 Images	0%	0%	0%				

Table 8: Training results with best 5 IQMs.

The results listed in Table 8, are obtained with the use of 5 IQMs, there were 50 real and 50 fake fingerprint images inputted to the system to train, the detailed results of the 100 samples of fingerprint using 6 classifiers are shown in Appendix C-5, Figures 62-67.

Best 5 Iqms: SME, PSNR, SNR, GPE, AND RAMD						
Classifier/Sample No.	FFR	FGR	HTER			
LDA/1000 Tested Fingerprint Images	10.4%	14.6%	12.5%			
QDA/1000 Images	13.8%	9.2%	11.5%			
Linear SVM/1000 Images	8.8%	6.4%	7.6%			
Quadratic SVM1000 Images	7.8%	6.2%	7%			
Logistic Regression/1000 Ima+Ges	6.2%	6.8%	6.5%			
Fine KNN/1000 Images	6.4%	8.6%	7.5%			

Table 9: Classification results with best 5 IQMs.

Table 9, shows the classification results with 5 IQMs for 500 real and 500 fake fingerprint images inputted to the system to classify, the detailed results of the 1000 samples of fingerprints tested for classification are shown in Appendix C-6, Figures 68-73. 4.3.4.

## 4.5.4 Experiments on The RFDS Training and Classification With 25 IQMs

<b>25 IQMs</b> SME, PSNR, SNR, GPE, RAMD, MD, TCD, NAE, SPE, SC, TED, NK, MSE, GME, AD, LMSE, MAS, MAMS, HLFREQIQ, BIQI, SSIM, RRED, JPEG, VIF, AND NIQE							
Classifier/Sample No.	FFR	FGR	HTER				
LDA/100 Images	4%	20%	12%				
QDA/100 Images	2%	0%	1%				
Linear SVM/100 Images	0%	2%	1%				
Quadratic SVM/100 Images	0%	0%	0%				
Logistic Regression/100 Images	0%	0%	0%				
Fine KNN/100 Images	0%	0%	0%				

Table 10: Training results with 25 IQMs.

The results listed in Table 10 are generated from ATV database for 50 real and 50 fake fingerprint images inputted to the system to train. The detailed results of the 100 samples of fingerprints using 25 IQMs using 6 classifiers are shown in Appendix C-7, Figures 74 -78.

<b>25 IQMs</b> SME, PSNR, SNR, GPE, RAMD, MD, TCD, NAE, SPE, SC, TED, NK, MSE, GME, AD, LMSE, MAS, MAMS, HLFREQIQ, BIQI, SSIM, RRED, JPEG, VIF, AND NIQE							
Classifier/Sample No.	FFR	FGR	HTER				
LDA/1000 Images	5.4%	0%	2.7%				
QDA/1000 Images	19%	0%	9.5%				
Linear SVM/1000 Images	2.6%	0%	1.3%				
Quadratic SVM/1000 Images	0.6%	0%	0.3%				
Logistic Regression/1000 Images	1%	0%	0.5%				
Fine KNN/1000 Images	1.2%	0%	0.6%				

Table 11: Classification results with 25 IQM's.

Table 11, shows the classification results with 25 IQM's for 500 real and 500 fake fingerprint images inputted to the system to classify. The detailed results of the 1000 samples of fingerprints for classification are shown in Appendix C-8, Figures 79-84.

4.5.5 Comparsion of the RFDS Performance With 5, 10, 15, And 25 Iqms Used

	<b>Result of Comparison For 5, 10, 15 And 25 IQMs</b>								
	Classifiers								
RFDS	FINEKNN	QDA	LDA	QSVM	LSVM	LOGREG			
25	N/A	12.8%	N/A	N/A	N/A	N/A			
25 IQMs	0.6%	9.5%	2.7%	0.3%	1.3%	0.5%			
15 IQMs	8.1%	14.8%	13.5%	5.8%	8%	8.5%			
10 IQMs	8.5%	13.5%	13.9%	6.2%	8.1%	13.4%			
5 IQMs	7.5%	11.5%	12.5%	7%	7.6%	6.5%			

Table 12: Result of comparison HTER (%) between our RFDS and RFDS [6].

Table 11, shows the comparison between RFDS performances with the usage of 25, 15, 10, 5 IQMs. We can clearly see that performance of our RFDS with 25 IQMs has better performance than the rest of the RFDS. Table 11, also shows that 10 and 15 IQMs RFDS are very similar to each other, their hter results are very close not like the RFDS with 5 IQMs where the result of its HTER has minmum 12% gap from the other RFDS. From Table 11, we can say that the best performance from the RFDS in the list is the one with 25 IQMs, because it's HTER score is the minimum one with 0.3%, and the worst RFDS is the one with the 5 IQMs which has the maximum HTER score with 29.7%.

# 4.6 Summary

We show the experiment setup of our RFDS, we also compared four of our RFDS using 5, 10, 15, and 25 IQMs with RFDS [6] which has used 25 IQMS. Our RFDS experiment was implemented using atv database [33], 100 fingerprint images were used for training process, and 1000 fingerprints were used for detection process. With the exclusion of our RFDS that used 5 IQMs, we can summarize that with less IQMs than RFDS [6], our system still managed to give us better results, listed all in table 11.

# **Chapter 5**

# CONCLUSION

In our thesis we analyzed similar to the "State-of-The-Art" RFDS systems; some of them used different image quality measures (IQMs), and also different types of classification process. The results of these RFDS systems in detecting fake fingerprint images still had errors and were not 100% accurate. These existing errors of detecting fingerprint image made us interested in implementing a RFDS system with high performance and fewer errors. In our studies we explained step by step procedure of how to implement our RFDS, implementation of Gaussian filter to the input image, implementation of the IQMs, implementation of the classification process, and implementation of the method of defining the best IQMs. All these implementations were done using MATLAB r2015b. We also show our experimental setup used for investigation of our RFDS. We compared our RFDS (using 5,10, 15, and 25 IQMs) with state of the art RFDS (using 25 IQMs). Our best RFDS hter result is 0.3% using 25 IQMs, and state oft he art RFDS hter was 12.8% using also the same 25 IQMs, which indicates that our RFDS performance is overall much better than the state of the art" RFDS even though in some experiments we used a smaller number of IQMs in our system compared to the state of the art RFDS.

## REFERENCES

- [1] Omran, S. S., & Salih, M. A. (2014). Design and Implementation of Multi-Model Biometric Identification System. *International Journal of Computer Applications*, 99(15), 14-21.
- [2] Mudholkar, S. S., Shende, P. M., & Sarode, M. V. (2012). Biometrics authentication technique for intrusion detection systems using fingerprint recognition. *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)*, 2(1), 57-65.
- [3] Srivastava, H. (2013). A comparison based study on biometrics for human recognition. *IOSR Journal of Computer Engineering (IOSR-JCE)*, *15*(1), 22-29.
- [4] Image Quality. (n.d.). In Wikipedia. Retrieved October 28, 2019, from https://en.wikipedia.org/wiki/Image\_quality#image\_quality\_metrics.
- [5] Naim, N. F., Yassin, A. I. M., Zamri, W. M. A. W., & Sarnin, S. S. (2011, March).
   Mysql Database for storage of fingerprint data. In 2011 UkSim 13th International Conference on Computer Modelling and Simulation (pp. 293-298). IEEE.
- [6] Galbally, J., Marcel, S., & Fierrez, J. (2013). Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition. *IEEE transactions on image processing*, 23(2), 710-724.

- [7] I. Avcibaş, I., Sankur, B., & Sayood, K. (2002). Statistical evaluation of image quality measures. *Journal of Electronic imaging*, 11(2), 206-223.
- [8] Huynh-Thu, Q., & Ghanbari, M. (2008). Scope of validity of PSNR in image/video quality assessment. *Electronics letters*, 44(13), 800-801.
- [9] Peak signal-to-noise ratio. In Wikipedia. Retrieved October 7, 2019, from https://en.wikipedia.org/wiki/Peak\_signal-to-noise\_ratio.
- [10] Yao, S., Lin, W., Ong, E., & Lu, Z. (2005, September). Contrast signal-to-noise ratio for image quality assessment. In *IEEE International Conference on Image Processing 2005* (Vol. 1, pp. I-397). IEEE.
- [11] Eskicioglu, A. M., & Fisher, P. S. (1995). Image quality measures and their performance. *IEEE Transactions on communications*, 43(12), 2959-2965.
- [12] Martini, M. G., Hewage, C. T., & Villarini, B. (2012). Image quality assessment based on edge preservation. *Signal Processing: Image Communication*, 27(8), 875-882.
- [13] Fourier Transform. (n.d.). In American Psychological Association. Retrieved October 27, 2019, from https://blog.apastyle.org/apastyle/urls/.
- [14] Nill, N. B., & Bouzas, B. (1992). Objective image quality measure derived from digital image power spectra. *Optical engineering*, 31(4), 813-826.

- [15] Liu, A., Lin, W., & Narwaria, M. (2011). Image quality assessment based on gradient similarity. *IEEE Transactions on Image Processing*, 21(4), 1500-1512.
- [16] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4), 600-612.
- [17] IQM Code. (n.d.). In Laboratory for Image & Video Engineering. Retrieved October 12, 2019, from http://live.ece.utexas.edu/research/quality/index.htm.
- [18] Sheikh, H. R., & Bovik, A. C. (2006). Image information and visual quality. *IEEE Transactions on image processing*, 15(2), 430-444.
- [19] Soundararajan, R., & Bovik, A. C. (2011). RRED indices: Reduced reference entropic differencing for image quality assessment. *IEEE Transactions on Image Processing*, 21(2), 517-526.
- [20] Zhu, X., & Milanfar, P. (2009, July). A no-reference sharpness metric sensitive to blur and noise. In 2009 International Workshop on Quality of Multimedia Experience (pp. 64-69). IEEE.
- [21] Wang, Z., Sheikh, H. R., & Bovik, A. C. (2002, September). No-reference perceptual quality assessment of JPEG compressed images. In *Proceedings*. *International Conference on Image Processing* (Vol. 1, pp. I-I). IEEE.

- [22] Moorthy, A. K., & Bovik, A. C. (2010). A two-step framework for constructing blind image quality indices. *IEEE Signal processing letters*, 17(5), 513-516.
- [23] Mittal, A., Soundararajan, R., & Bovik, A. C. (2012). Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3), 209-212.
- [24] Linear discriminant analysis. In Wikipedia. Retrieved October 10, 2019, from https://en.wikipedia.org/wiki/Linear\_discriminant\_analysis.
- [25] Quadratic classifier. In Wikipedia. Retrieved October 5, 2019, from https://en.wikipedia.org/wiki/Quadratic\_classifier.
- [26] LivDet Databases. (n.d.). In Liveness Detection Competition. Retrieved October 20, 2019, from http://livdet.org/registration.php
- [27] Galbally, J., Alonso-Fernandez, F., Fierrez, J., & Ortega-Garcia, J. (2009, September). Fingerprint liveness detection based on quality measures. In 2009 *First IEEE International Conference on Biometrics, Identity and Security* (*BIdS*) (pp. 1-8). IEEE.
- [28] Galbally, J., Alonso-Fernandez, F., Fierrez, J., & Ortega-Garcia, J. (2012). A high performance fingerprint liveness detection method based on quality related features. *Future Generation Computer Systems*, 28(1), 311-321.

- [29] Lourde, M., & Khosla, D. (2010). Fingerprint Identification in Biometric SecuritySystems. International Journal of Computer and Electrical Engineering, 2(5), 852.
- [30] Arunalatha, G., & Ezhilarasan, M. (2015). Fingerprint spoof detection using quality features. Int. J. Secur. Its Appl, 9(10), 83-94.
- [31] Mohammed, M. O. (2017). Parametric Real Face Images Detection System (RFIDS) Using Multiple Classifiers (Master's thesis, Eastern Mediterranean University (EMU)-Doğu Akdeniz Üniversitesi (DAÜ)).
- [32] Gaussian Filter. (n.d.). In Wikipedia. Retrieved October 14, 2019, from https://en.wikipedia.org/wiki/Gaussian\_filter.
- [ 33] DESCRIPTION OF ATVS-FFp DB. (n.d.). In Biometrics Ideal Test. Retrieved October 1, 2019, from http://biometrics.idealtest.org/dbDetailForUser.do?id=11

APPENDICES

## **Appendix A: Feature Extraction Code:**

#### **1. MSE Function**

Function Iqm\_Mse = Mean\_Sqr\_Error(Rel\_Img, Enhcd\_Img)

Rel\_Img = Double(Rel\_Img);

Enhcd\_Img = Double(Enhcd\_Img);

[M N] = Size(Rel\_Img);

Eror = Rel\_Img - Enhcd\_Img;

Iqm\_Mse = Sum(Sum(Eror .\* Eror)) / (M \* N);

## 2. PSNR Function

Function Iqm\_Psnr = Pk\_Signalt\_Nse\_Ratio(Rel\_Img, Enhcd\_Img)

Rel\_Img = Double(Rel\_Img);

Enhcd\_Img = Double(Enhcd\_Img);

[M N] = Size(Rel\_Img);

Eror = Rel\_Img - Enhcd\_Img;

Iqm\_Mse = Sum(Sum(Eror .\* Eror)) / (M \* N);

If  $(Iqm_Mse > 0)$ 

Iqm\_Psnr = 10\*Log10((255.\*255)/Iqm\_Mse);

Else

End

#### **3. SNR Function**

Function Iqm\_Snr = Signal\_Nse\_Ratio(Rel\_Img, Enhcd\_Img)

Rel\_Img = Double(Rel\_Img);

Enhcd\_Img = Double(Enhcd\_Img);

[M, N] = Size(Rel\_Img);

Eror = Rel\_Img - Enhcd\_Img;

Iqm\_Mse = Sum(Sum(Eror .\* Eror)) / (M \* N); If(Iqm\_Mse > 0)

Iqm\_Snr = 10\*Log10((Sum(Sum(Rel\_Img.\*Rel\_Img)))./(M .\* N .\* Iqm\_Mse)); Else

Iqm\_Snr = 99; End

#### 4. SC Function

Function Iqm\_Sc = Struct\_Cntnt(Rel\_Img, Enhcd\_Img)

Rel\_Img = Double(Rel\_Img);

Enhcd\_Img = Double(Enhcd\_Img);

Iqm\_Sc = Sum(Sum(Rel\_Img .\* Rel\_Img)) ./ Sum(Sum(Enhcd\_Img .\* Enhcd\_Img));

#### 5. MD Function

Function Iqm\_Md = Max\_Diff(Rel\_Img, Enhcd\_Img)

Rel\_Img = Double(Rel\_Img);

Enhcd\_Img = Double(Enhcd\_Img);

Eror = Rel\_Img - Enhcd\_Img;

Iqm\_Md = Max(Max(Abs(Eror)));

#### **6.AD** Function

Function Ad = Avrge\_Diff(Rel\_Img, Enhcd\_Img)

Rel\_Img = Double(Rel\_Img);

Enhcd\_Img = Double(Enhcd\_Img);

[M N] = Size(Rel\_Img);

Eror = Rel\_Img - Enhcd\_Img;

Iqm\_Ad = Sum(Sum(Eror)) / (M \* N);

#### 7. NAE Function

Function Iqm\_Nae = Normalized\_Abs\_Error(Rel\_Img, Enhcd\_Img)

Rel\_Img = Double(Rel\_Img);

Enhcd\_Img = Double(Enhcd\_Img);

Eror = Rel\_Img - Enhcd\_Img;

Iqm\_Nae = Sum(Sum(Abs(Eror))) / Sum(Sum(Abs(Rel\_Img)));

#### 8. RAMD Function

Function Iqm\_Ramd = Ravg\_Md(Rel\_Img, Enhcd\_Img)

Rel\_Img = Double(Rel\_Img);

Enhcd\_Img = Double(Enhcd\_Img);

Eror = Rel\_Img - Enhcd\_Img;

Eror=Abs(Eror);

A1=Eror(:);

B=Flipud(Unique(Sort(A1)));

Resultatt=B(1:10);

Resultat\_1=Resultatt(1:10,:)

R = 10;

Iqm\_Ramd = Sum((Abs(Resultat\_1)))/R;

#### 9. NCC Function

Function Iqm\_Ncc = Normalized\_Crs\_Correlation(Rel\_Img, Enhcd\_Img)

Rel\_Img = Double(Rel\_Img);

Enhcd\_Img = Double(Enhcd\_Img);

Iqm\_Ncc = Sum(Sum(Rel\_Img .\* Enhcd\_Img)) ./ Sum(Sum(Rel\_Img .\* Rel\_Img));

#### **10. TED Function**

Function Iqm\_Ted = Total\_Edg\_Dif(Rel\_Img, Enhcd\_Img)

Rel\_Img=Edge(Rel\_Img,'Sobel');

Figure, Imshow(Rel\_Img)

Rel\_Img = Double(Rel\_Img);

Enhcd\_Img=Edge(Enhcd\_Img,'Sobel');

Figure, Imshow(Rel\_Img)

Enhcd\_Img = Double(Enhcd\_Img);

[M, N] = Size(Rel\_Img);

Eror = Rel\_Img - Enhcd\_Img;

Iqm\_Ted = Sum(Sum(Abs(Eror))) / (M \* N);

#### **11.TCD Function**

Function Iqm\_Tcd = Total\_Crner\_Diff(Rel\_Img, Enhcd\_Img)

Rel\_Img = Double(Rel\_Img);

Enhcd\_Img = Double(Enhcd\_Img);

Ncr\_Orig = Detectharrisfeatures(Rel\_Img);

Imshow(Rel\_Img);

Hold On;

Plot(Ncr\_Orig);

Ncr\_Orig = Length(Ncr\_Orig);

Ncr\_Dist = Detectharrisfeatures(Enhcd\_Img); Imshow(Enhcd\_Img); Hold On;

Plot(Ncr\_Dist);

Ncr\_Dist = Length(Ncr\_Dist);

Max\_1= Max(Ncr\_Orig,Ncr\_Dist);

Iqm\_Tcd =(Abs(Ncr\_Orig-Ncr\_Dist))/Max\_1;

## **12. SPE Function**

Function Iqm\_Spe = Spectral\_Ph\_Error(Rel\_Img, Enhcd\_Img)

Rel\_Img = Double(Rel\_Img);

Enhcd\_Img = Double(Enhcd\_Img);

Fft\_A = Fft2(Double(Rel\_Img));

Argfft\_A =Angle(Fft\_A);

Fft\_B = Fft2(Double(Enhcd\_Img));

Argfft\_B =Angle(Fft\_B);

 $[M, N] = Size(Rel_Img);$ 

Eror=Abs(Argfft\_A)-Abs(Argfft\_B);

Iqm\_Spe =Sum(Sum( Eror.\*Eror)) / (M \* N);

#### **13. SME Function**

Function Iqm\_Sme = Spectral\_Mg\_Error(Rel\_Img, Enhcd\_Img)

Rel\_Img = Double(Rel\_Img);

Enhcd\_Img = Double(Enhcd\_Img);

Fft\_A = Fft2(Double(Rel\_Img));

Z1rr=Real(Fft\_A);

Z1ii=Imag(Fft\_A);

Fft\_A1=Sqrt((Z1rr.\*Z1rr)+(Z1ii.\*Z1ii));

Fft\_B = Fft2(Double(Enhcd\_Img));

Z2rr=Real(Fft\_B);

Z2ii=Imag(Fft\_B);

Fft\_B1=Sqrt((Z2rr.\*Z2rr)+(Z2ii.\*Z2ii));

[M, N] = Size(Rel\_Img); Eror=Abs(Fft\_A1)-Abs(Fft\_B1);

Iqm\_Sme =Sum(Sum( Eror.\*Eror)) / (M \* N);

## **14. GPE Function**

Function Iqm\_Gpe = Gradient\_Ph\_Eror(Rel\_Img, Enhcd\_Img)

Rel\_Img = Double(Rel\_Img);

Enhcd\_Img = Double(Enhcd\_Img);

[Fx1,Fy1] = Gradient(Rel\_Img);

Zz=Complex(Fx1,Fy1);

Arggmagorig = Angle(Zz);

[Fx1,Fy1] = Gradient(Enhcd\_Img);

Yy=Complex(Fx1,Fy1);

Arggma\_Dist = Angle(Yy);

 $[M, N] = Size(Rel_Img);$ 

Eror=Abs(Arggmagorig)-Abs(Arggma\_Dist);

Iqm\_Gpe =Sum(Sum( Eror.\*Eror)) / (M \* N)

## **15. GME Function**

Function Iqm\_Gme = Gradient\_Mg\_Error(Rel\_Img, Enhcd\_Img)

Rel\_Img = Double(Rel\_Img);

Enhcd\_Img = Double(Enhcd\_Img);

[Fx1,Fy1] = Gradient(Rel\_Img);

Gmgorig=Sqrt((Fx1.\*Fx1)+(Fy1.\*Fy1)); [Fx1,Fy1] = Gradient(Enhcd\_Img);

Gmgdis=Sqrt((Fx1.\*Fx1)+(Fy1.\*Fy1));

[M, N] = Size(Rel\_Img);

Eror=Abs(Gmgorig)-Abs(Gmgdis);

Eror=Eror.\*Eror;

Iqm\_Gme =Sum(Sum( Eror)) / (M \* N)

### **16. MAS Function**

Function Mas= Meananglesimilarity(Realimg, Enhimg)%Mas

[M N]= Size(Realimg);

Alfa2= 2/180 .\* (Acosd(Dot(Realimg,Enhimg)/(Norm(Realimg).\* Norm(Enhimg))));

Angle2 = Sum( Alfa2 );

Mas = 1-Angle2/(M\*N);

End

#### **17. MAMS Function**

Function Mams= Meananglemagnitudesimilarity(Realimg, Enhimg)

[M N]= Size(Realimg);

Alfa= 2/Pi .\* (Acos(Dot(Realimg, Enhimg)/(Norm(Realimg).\* Norm(Enhimg))));

X= 1 - ((1-Alfa)\*(1 - (Norm(Realimg-Enhimg)/255)));

Sum = Sum(X);

Mams = Sum / (M\*N);

End

#### **18. LMSE Function**

Function Lmse= Laplacianmeansquarederror(Realing, Enhimg)%Lmse

[M N]= Size(Realimg);

Op=4\*Del2(Realimg);

Op2=4\*Del2(Enhimg);

Lmse=Sum(Sum((Op-Op2).^2))/Sum(Sum(Op.^2));

End

### **19. SSIM Function**

Function Ssim= Structuralsimilarityindex(Realimg, Ehnimg)
Img1=Double(Realimg);
Img2=Double(Ehnimg);
K = [0.01 0.03];
Window = Fspecial('Gaussian', 11, 1.5);
L = 255;
Ssim = Ssim Index(Img1, Img2, K, Window, L);

## **20. VIF Function**

Function Vif=Vifvec(Imorg,Imdist);

% ------Copyright Notice Starts With This Line-----

% Copyright (C) 2005 The University Of Texas At Austin

% All Rights Reserved.

% Permission Is Hereby Granted, Without Written Agreement And Without License

Or Royalty Fees, To Use, Copy,

% Modify, And Distribute This Code (The Source Files) And Its Documentation For

% Any Purpose, Provided That The Copyright Notice In Its Entirety Appear In All

Copies Of This Code, And The

% Original Source Of This Code, Laboratory For Image And Video Engineering (Live, Http://Live.Ece.Utexas.Edu)

% At The University Of Texas At Austin (Ut Austin,

% Http://Www.Utexas.Edu), Is Acknowledged In Any Publication That Reports Research Using This Code. The Research

% Is To Be Cited In The Bibliography As:

% H. R. Sheikh And A. C. Bovik, "Image Information And Visual Quality", Ieee Transactions On

% Image Processing, (To Appear).

% In No Event Shall The University Of Texas At Austin Be Liable To Any Party For Direct, Indirect, Special, Incidental,

% Or Consequential Damages Arising Out Of The Use Of This Database And Its Documentation, Even If The University Of Texas

% At Austin Has Been Advised Of The Possibility Of Such Damage.

% The University Of Texas At Austin Specifically Disclaims Any Warranties, Including, But Not Limited To, The Implied

% Warranties Of Merchantability And Fitness For A Particular Purpose. The Database Provided Hereunder Is On An "As Is" Basis,

% And The University Of Texas At Austin Has No Obligation To Provide Maintenance, Support, Updates, Enhancements, Or Modifications.

% ------Copyright Notice Ends With This Line-----

%This Is An Implementation Of The Algorithm For Calculating The

%Visual Information Fidelity (Vif) Measure (May Also Be Known As The Sheikh

%-Bovik Index) Between Two Images. Please Refer

%To The Following Paper:

%H. R. Sheikh And A. C. Bovik "Image Information And Visual Quality"
%Ieee Transactios On Image Processing, In Publication, May 2005.
%Download Manuscript Draft From Http://Live.Ece.Utexas.Edu In The
%Publications Link

%This Implementation Is Slightly Differnet From The One Used To Report %Results In The Paper Above. The Modification Have To Do With Using More %Subands Than Those Used In The Paper, Better Handling Of Image Boundaries, %And A Window That Automatically Resizes Itself Based On The Scale. %Report Bugfixes And Comments To Hamid.Sheikh@Ieee.Org

%-----

% Prerequisites: The Steerable Pyramid Toolbox. Available At

% Http://Www.Cns.Nyu.Edu/~Lcv/Software.Html

%Input : (1) Img1: The Reference Image

% (2) Img2: The Distorted Image (Order Is Important)

%Output: (1) Vif Te Visual Information Fidelity Measure Between The Two Images %Default Usage:

% Given 2 Test Images Img1 And Img2, Whose Dynamic Range Is 0-255

% Vif = Vifvec(Img1, Img2);

%Advanced Usage:

% Users May Want To Modify The Parameters In The Code.

% (1) Modify Sigma\_Nsq To Find Tune For Your Image Dataset.

- % (2) Mxm Is The Block Size That Denotes The Size Of A Vector Used In The
- % Gsm Model.
- % (3) Subbands Included In The Computation

%====

M=3;

Subbands=[4 7 10 13 16 19 22 25];

Sigma\_Nsq=0.4;

% Do Wavelet Decomposition. This Requires The Steerable Pyramid. You Can

% Use Your Own Wavelet As Long As The Cell Arrays Org And Dist Contain

% Corresponding Subbands From The Reference And The Distorted Images

% Respectively.

[Pyr,Pind] = Buildspyr(Imorg, 4, 'Sp5filters', 'Reflect1'); % Compute Transform

Org=Ind2wtree(Pyr,Pind); % Convert To Cell Array

[Pyr,Pind] = Buildspyr(Imdist, 4, 'Sp5filters', 'Reflect1');

Dist=Ind2wtree(Pyr,Pind);

% Calculate The Parameters Of The Distortion Channel

[G\_All,Vv\_All]=Vifsub\_Est\_M(Org,Dist,Subbands,M);

% Calculate The Parameters Of The Reference Image

[Ssarr, Larr, Cuarr]=Refparams\_Vecgsm(Org,Subbands,M);

% Reorder Subbands. This Is Needed Since The Outputs Of The Above Functions

% Are Not In The Same Order

Vvtemp=Cell(1,Max(Subbands));

Ggtemp=Vvtemp;

For(Kk=1:Length(Subbands))

Vvtemp{Subbands(Kk)}=Vv\_All{Kk};

Ggtemp{Subbands(Kk)}=G\_All{Kk};

End

% Compute Reference And Distorted Image Information From Each Subband

For I=1:Length(Subbands)

Sub=Subbands(I);

G=Ggtemp{Sub};

Vv=Vvtemp{Sub};

Ss=Ssarr{Sub};

Lambda = Larr(Sub,:);,

Cu=Cuarr{Sub};

% How Many Eigenvalues To Sum Over. Default Is All.

Neigvals=Length(Lambda);

% Compute The Size Of The Window Used In The Distortion Channel Estimation,

And Use It To Calculate The Offset From Subband Borders

% We Do This To Avoid All Coefficients That May Suffer From Boundary

% Effects

Lev=Ceil((Sub-1)/6);

Winsize=2^Lev+1; Offset=(Winsize-1)/2;

Offset=Ceil(Offset/M);

% Select Only Valid Portion Of The Output.

G=G(Offset+1:End-Offset,Offset+1:End-Offset);

Vv=Vv(Offset+1:End-Offset,Offset+1:End-Offset);

Ss=Ss(Offset+1:End-Offset,Offset+1:End-Offset);

%Vif

Temp1=0; Temp2=0;

For J=1:Length(Lambda)

Temp1=Temp1+Sum(Sum((Log2(1+G.\*G.\*Ss.\*Lambda(J)./(Vv+Sigma\_Nsq))))); %

Distorted Image Information For The I'th Subband

Temp2=Temp2+Sum(Sum((Log2(1+Ss.\*Lambda(J)./(Sigma\_Nsq))))); % Reference Image Information End Num(I)=Temp1; Den(I)=Temp2; End % Compuate Vif Vif=Sum(Num)./Sum(Den);

## **21. RRED Function**

Function Rred=Rred\_Out(Realing, Enhimg)

% Total = 5;

Original=Realimg;

Fake= Enhimg;

Rredoutput= Extract\_Info(Original, Fake);

[X Y] = Size(Rredoutput);

Sumoutput=0;

For I=1 :X

For J=1:Y

Sumoutput = Sumoutput + Rredoutput(I,J);

End

End

Rred=Norm(Sumoutput);

% End

### **22.HLFI Function**

Function Hlfreqiq= High\_Low\_Freq\_Index(Realimg)

[R,C]=Size(Realimg);

N = C;

Imgfft =Fft(Fft2(Realimg));

Colhalf = (Round(N / 2));

Freqsel = 0.15;

Freqcol = Round(Freqsel \* N);

Lowfreqcolhalf = (Round(Freqcol / 2.0));

Fftres = Imgfft; %R# Np.Fft.Fft2(Image)

Fftmag = Abs(Fftres);

Totalenergy = Sum(Sum(Fftmag));

%# Print(Totalenergy)

Lowidx = Colhalf - Lowfreqcolhalf;

Hiidx = Colhalf + Lowfreqcolhalf;

Lowfreqmag = Fftmag(:, Lowidx:Hiidx);

Lowfreqmagtotal = Sum(Sum(Lowfreqmag));

Fftmag(:, Lowidx:Hiidx) = 0;

Highfreqmagtotal =Sum( Sum(Fftmag));

Hlfreqiq = Abs(Lowfreqmagtotal - Highfreqmagtotal) /(Totalenergy);

End

## 23. JQI Function

Function [Score B A Z] = Jpeg\_Quality\_Score(Img)

%Copyright (C) 2002 The University Of Texas At Austin

%All Rights Reserved.

%This Program Is Free Software; You Can Redistribute It And/Or Modify

%It Under The Terms Of The Gnu General Public License As Published By

RFDS training and detection structure %(At Your Option) Any Later Version.

%This Program Is Distributed In The Hope That It Will Be Useful,

%But Without Any Warranty; Without Even The Implied Warranty Of

%Merchantability Or Fitness For A Particular Purpose. See The

%Gnu General Public License For More Details.

%The Gnu Public License Is Available In The File License, Or You

%Can Write To The Free Software Foundation, Inc., 59 Temple Place -

%Suite 330, Boston, Ma 02111-1307, Usa, Or You Can Find It On The

%World Wide Web At Http://Www.Fsf.Org.

% Author : Zhou Wang

% Version : 1.0

% Modified: Anush Krishna Moorthy

% Version: 1.Buqi

%The Authors Are With The Laboratory For Image And Video Engineering

%(Live), Department Of Electrical And Computer Engineering, The

%University Of Texas At Austin, Austin, Tx.

%

%Kindly Report Any Suggestions Or Corrections To Zhouwang@Ieee.Org

%This Is An Implementation Of The Algorithm For Calculating The Quality %Score Of Jpeg Compressed Images Proposed By Zhou Wang, Hamid R. Sheikh %And Alan C. Bovik. Please Refer To The Paper: Zhou Wang, Hamid R. Sheikh %And Alan C. Bovik, "No-Reference Perceptual Quality Assessment Of Jpeg %Compressed Images," Submitted To Ieee International Conference On Image %Processing, Sept. 2002.

%You Can Change This Program As You Like And Use It Anywhere, But Please %Refer To Its Original Source (Cite Our Paper And Our Web Page At %Http://Anchovy.Ece.Utexas.Edu/~Zwang/Research/Nr\_Jpeg\_Quality/Index.Html) %

%Input : A Test 8bits/Pixel Grayscale Image Loaded In A 2-D Array

%Output: A Quality Score Of The Image. The Score Typically Has A Value

% Between 1 And 10 (10 Represents The Best Quality, 1 The Worst).%Usage:

%1. Load The Image, For Example

% Image = Imread('Testimage.Jpg');

%2. Call This Function To Calculate The Quality Score:

% Quality\_Score = Jpeg\_Quality\_Score(Image)

%\_\_\_\_\_\_

If (Nargin > 1)

Score = -1;

Return;

End

[M N] = Size(Img);

If (M < 16 | N < 16)

Score = -2;

Return;

End

X = Double(Img);

% Feature Extraction:

% 1. Horizontal Features

 $D_H = X(:, 2:N) - X(:, 1:(N-1));$ 

 $B_H = Mean2(Abs(D_H(:, 8:8:8*(Floor(N/8)-1))));$ 

 $A_H = (8*Mean2(Abs(D_H)) - B_H)/7;$ 

 $Sig_H = Sign(D_H);$ 

Left\_Sig = Sig\_H(:, 1:(N-2));

 $Right_Sig = Sig_H(:, 2:(N-1));$ 

Z\_H = Mean2((Left\_Sig.\*Right\_Sig)<0);</pre>

% 2. Vertical Features

 $D_V = X(2:M, :) - X(1:(M-1), :);$ 

 $B_V = Mean2(Abs(D_V(8:8:8*(Floor(M/8)-1), :)));$ 

 $A_V = (8*Mean2(Abs(D_V)) - B_V)/7;$ 

 $Sig_V = Sign(D_V);$ 

Up\_Sig = Sig\_V(1:(M-2), :);

Down\_Sig = Sig\_V(2:(M-1), :);

Z\_V = Mean2((Up\_Sig.\*Down\_Sig)<0);</pre>

% 3. Combined Features

 $B = (B_H + B_V)/2;$ 

 $A = (A_H + A_V)/2;$   $Z = (Z_H + Z_V)/2;$ % Quality Prediction Alpha = -927.4240; Beta = 850.8986;Gamma1 = 235.4451 ;Gamma2 = 128.7548;Gamma3 = -341.4790; Score = Abs(Alpha + Beta\*(B.^(Gamma1/10000))\*(A.^(Gamma2/10000))\*(Z.^(Gamma3/10000)));

#### 24. BIQI Function

Function [Score B A Z] = Blinimagequalityindex(Rimg)

%\_\_\_\_\_

%Copyright (C) 2002 The University Of Texas At Austin

%All Rights Reserved.

%This Program Is Free Software; You Can Redistribute It And/Or Modify

%It Under The Terms Of The Gnu General Public License As Published By

%The Free Software Foundation; Either Version 2 Of The License, Or

%(At Your Option) Any Later Version.

%

%This Program Is Distributed In The Hope That It Will Be Useful,

%But Without Any Warranty; Without Even The Implied Warranty Of

%Merchantability Or Fitness For A Particular Purpose. See The

%Gnu General Public License For More Details.

%

%The Gnu Public License Is Available In The File License, Or You

%Can Write To The Free Software Foundation, Inc., 59 Temple Place -%Suite 330, Boston, Ma 02111-1307, Usa, Or You Can Find It On The %World Wide Web At Http://Www.Fsf.Org. %

% Author : Zhou Wang

% Version : 1.0

% Modified: Anush Krishna Moorthy

% Version: 1.Buqi

%

%The Authors Are With The Laboratory For Image And Video Engineering %(Live), Department Of Electrical And Computer Engineering, The %University Of Texas At Austin, Austin, Tx.

%

%Kindly Report Any Suggestions Or Corrections To Zhouwang@Ieee.Org %

%This Is An Implementation Of The Algorithm For Calculating The Quality %Score Of Jpeg Compressed Images Proposed By Zhou Wang, Hamid R. Sheikh %And Alan C. Bovik. Please Refer To The Paper: Zhou Wang, Hamid R. Sheikh %And Alan C. Bovik, "No-Reference Perceptual Quality Assessment Of Jpeg %Compressed Images," Submitted To Ieee International Conference On Image %Processing, Sept. 2002.

%

%You Can Change This Program As You Like And Use It Anywhere, But Please %Refer To Its Original Source (Cite Our Paper And Our Web Page At %Http://Anchovy.Ece.Utexas.Edu/~Zwang/Research/Nr\_Jpeg\_Quality/Index.Html)
%Input : A Test 8bits/Pixel Grayscale Image Loaded In A 2-D Array
%Output: A Quality Score Of The Image. The Score Typically Has A Value
% Between 1 And 10 (10 Represents The Best Quality, 1 The Worst).

%Usage:

%1. Load The Image, For Example

% Image = Imread('Testimage.Jpg');

%2. Call This Function To Calculate The Quality Score:

% Quality\_Score = Blinimagequalityindex(Image)

If (Nargin > 1)

Score = -1;

Return;

End

[M N] = Size(Rimg);

If (M < 16 | N < 16)

Score = -2;

Return;

End

X = Double(Rimg);

% Feature Extraction:

% 1. Horizontal Features

 $D_H = X(:, 2:N) - X(:, 1:(N-1));$ 

 $B_H = Mean2(Abs(D_H(:, 8:8:8*(Floor(N/8)-1))));$ 

 $A_H = (8*Mean2(Abs(D_H)) - B_H)/7;$ 

 $Sig_H = Sign(D_H);$ 

Left\_Sig = Sig\_H(:, 1:(N-2));

Right\_Sig = Sig\_H(:, 2:(N-1));

Z\_H = Mean2((Left\_Sig.\*Right\_Sig)<0);

% 2. Vertical Features

 $D_V = X(2:M, :) - X(1:(M-1), :);$ 

 $B_V = Mean2(Abs(D_V(8:8:8*(Floor(M/8)-1), :)));$ 

 $A_V = (8*Mean2(Abs(D_V)) - B_V)/7;$ 

 $Sig_V = Sign(D_V);$ 

 $Up_Sig = Sig_V(1:(M-2), :);$ 

Down\_Sig = Sig\_V(2:(M-1), :);

Z\_V = Mean2((Up\_Sig.\*Down\_Sig)<0);</pre>

% 3. Combined Features

 $\mathbf{B} = (\mathbf{B}_{H} + \mathbf{B}_{V})/2;$ 

 $A = (A_H + A_V)/2;$ 

 $Z = (Z_H + Z_V)/2;$ 

% Quality Prediction

Alpha = -927.4240; Beta = 850.8986;Gamma1 = 235.4451 ;Gamma2 =

128.7548;Gamma3 =-341.4790;

Score = Abs(Alpha + Beta\*(B.^(Gamma1/10000))\*(A.^(Gamma2/10000))\*(Z.^(Gamma3/ 10000))); NIQE Function Function Quality = Computequality(Im,Blocksizerow,Blocksizecol,...

Blockrowoverlap,Blockcoloverlap,Mu Prisparam,Cov Prisparam)

% Input

% Im - Image Whose Quality Needs To Be Computed

% Blocksizerow - Height Of The Blocks In To Which Image Is Divided

% Blocksizecol - Width Of The Blocks In To Which Image Is Divided

% Blockrowoverlap - Amount Of Vertical Overlap Between Blocks

% Blockcoloverlap - Amount Of Horizontal Overlap Between Blocks

% Mu\_Prisparam - Mean Of Multivariate Gaussian Model

% Cov\_Prisparam - Covariance Of Multivariate Gaussian Model

% For Good Performance, It Is Advisable To Use Make The Multivariate Gaussian Model

% Using Same Size Patches As The Distorted Image Is Divided In To

% Output

%Quality - Quality Of The Input Distorted Image

% Example Call

%Quality = Computequality(Im,96,96,0,0,Mu\_Prisparam,Cov\_Prisparam)

% -----

%Number Of Features

% 18 Features At Each Scale

Featnum = 18;

°/<sub>0</sub>------

%Compute Features

If(Size(Im,3)==3)

Im = Rgb2gray(Im);

End

Im = Double(Im);

[Row Col] = Size(Im);

Block\_Rownum = Floor(Row/Blocksizerow);

Block\_Colnum = Floor(Col/Blocksizecol);

Im = Im(1:Block\_Rownum\*Blocksizerow,1:Block\_Colnum\*Blocksizecol);

[Row Col] = Size(Im);

Block\_Rownum = Floor(Row/Blocksizerow);

Block\_Colnum = Floor(Col/Blocksizecol);

Im = Im(1:Block\_Rownum\*Blocksizerow, .

1:Block\_Colnum\*Blocksizecol);

Window = Fspecial('Gaussian',7,7/6);

Window = Window/Sum(Sum(Window));

Scalenum = 2;

Warning('Off')

Feat = [];

For Itr\_Scale = 1:Scalenum

Mu = Imfilter(Im,Window,'Replicate');

 $Mu_Sq = Mu.*Mu;$ 

Sigma = Sqrt(Abs(Imfilter(Im.\*Im,Window,'Replicate') - Mu\_Sq));

Structdis = (Im-Mu)./(Sigma+1);

Feat\_Scale = Blkproc(Structdis,[Blocksizerow/Itr\_Scale Blocksizecol/Itr\_Scale],

[Blockrowoverlap/Itr\_Scale Blockcoloverlap/Itr\_Scale], ...

@Computefeature);

Feat\_Scale = Reshape(Feat\_Scale,[Featnum ....

Size(Feat\_Scale,1)\*Size(Feat\_Scale,2)/Featnum]);

Feat\_Scale = Feat\_Scale';

```
If(Itr_Scale == 1)
```

Sharpness = Blkproc(Sigma,[Blocksizerow Blocksizecol], ...

[Blockrowoverlap Blockcoloverlap],@Computemean);

Sharpness = Sharpness(:);

End

Feat = [Feat Feat\_Scale];

Im =Imresize(Im,0.5);

End

% Fit A Mvg Model To Distorted Patch Features

Distparam = Feat;

Mu\_Distparam = Nanmean(Distparam);

Cov\_Distparam = Nancov(Distparam);

% Compute Quality

Invcov\_Param = Pinv((Cov\_Prisparam+Cov\_Distparam)/2);

Quality = Sqrt((Mu\_Prisparam-Mu\_Distparam)\* ...

Invcov\_Param\*(Mu\_Prisparam-Mu\_Distparam)');

## **Appendix B: Code for Ranking IQM's**

#### 1. Mean Squared Eror

 $R_Mse = Mse(1:500);$ 

 $F_Mse = Mse(501:1000);$ 

 $Avg_R_Mse = Mean(R_Mse);$ 

Disp('Average Of R\_Mse :');

Disp(Avg\_R\_Mse);

Avg\_F\_Mse = Mean(F\_Mse);

Disp('Average Of F\_Mse :');

Disp(Avg\_F\_Mse);

Disp('Average Of R\_Mse - Average Of F\_Mse :');

Dif\_Btwn\_Mse\_Fmse= Abs(Avg\_R\_Mse-Avg\_F\_Mse);

Disp(Dif\_Btwn\_Mse\_Fmse);

Disp('-----');

Note: Apply The Same Code For Every Iqm

#### 2. Ranking the Best 15 IQMs

Title={'Mse.'; 'Snr.'; 'Psnr.'; 'Nk.'; 'Ad.'; 'Sc.'; 'Md.'; 'Ramd.'; 'Nae.'; 'Ted.'; 'Spe.'; 'Sme.'; 'Gme.'; 'Gpe.'; 'Tcd.'};

Best15=[Dif\_Btwn\_Mse\_Fmse;Dif\_Btwn\_Snr\_Fsnr;Dif\_Btwn\_Psnr\_Fpsnr;Dif\_Btw n\_Nk\_Fnk;Dif\_Btwn\_Ad\_Fad;Dif\_Btwn\_Sc\_Fsc;Dif\_Btwn\_Md\_Fmd;Dif\_Btwn\_R amd\_Framd;Dif\_Btwn\_Nae\_Fnae;Dif\_Btwn\_Ted\_Fted;Dif\_Btwn\_Spe\_Fspe;Dif\_Bt wn\_Sme\_Fsme;Dif\_Btwn\_Gme\_Fgme;Dif\_Btwn\_Gpe\_Fgpe;Dif\_Btwn\_Tcd\_Ftcd]; Tbla = Table(Title,Best15) Disp('-----Best 15 Iqm Rank (Classification)-----');

 $Tblb = Sortrows(Tbla, \{'Best15'\}, \{'Descend'\})$ 

## 3. Calculating FGR, FFR, And HTER Of The RFDS

Yfit = Trainedclassifier1.Predictfcn(T) Truecount=0; Fakecount=0; Truecount1=0; Fakecount1=0; %Real Images Count Z=500; For B=1:Z; If Isequal(Yfit{B},'Real') Truecount=Truecount+1; Else Isequal(Yfit{B},'Fake') Fakecount=Fakecount+1; End End %500 Fake Images Count M=1000; For Q=501:M; If Isequal(Yfit{Q},'Real') Truecount1=Truecount1+1; Else Isequal(Yfit{Q},'Fake') Fakecount1=Fakecount1+1;

End

End

Disp('500 Real Fingerprint Images Result :');

Rr= ['Number Of Real Images Predict As Real = ',Num2str(Truecount)];

Rf= ['Number Of Real Images Predict As Fake = ',Num2str(Fakecount)];

Disp(Rr);

Disp(Rf);

Disp('500 Fake Fingerprint Images Result :');

Rr1= ['Number Of Fake Images Predict As Fake = ',Num2str(Fakecount1)];

Rf1= ['Number Of Fake Images Predict As Real = ',Num2str(Truecount1)];

Disp(Rr1);

Disp(Rf1);

Fgr=['Fals Genuine Rate(Fgr) = ',Num2str(Truecount1/500)];

Ffr=['Fals Fake Rate(Ffr) = ',Num2str(Fakecount/500)];

Hter=['HalfTotalErrorRate(Hter)=',Num2str(((Truecount1/500)+(Fakecount/500))/2)];Disp(Fgr);Disp(Ffr);

Disp(Hter);

## **Appendix C: Screenshots and Tables of Our Rfds Results**

## 1. Table and Screenshot of Training Results For 100 Fingerprint Images Using 15 IQMs

The following are the screenshots providing results of a dataset with 50 real and fake fingerprint images, with 15 image quality measures calculated for each fingerprint image. Table d.1 shows he result of 15 image quality measures of 50 real fingerprint images and 50 fake fingerprint images used for training. 100 fingerprint images with 15 IQMs results using 6 classifiers.

#### 1. LDA

Image       Image <th< th=""><th>tion 🔎</th></th<>	tion 🔎
<pre></pre>	
Current Folder     Image: Status - C.U.BerskLENOVCD-PCLDesktophp, New folder (2)Abdu codelTM_TEST_10.m       Marker A     TM_TEST_10m x Create_TM_10m x Create_TM_10m x bettlowdrm x +       Marker A     TM_TEST_10m x Create_TM_10m x Create_TM_10m x bettlowdrm x +       Marker A     Status - C.U.BerskLENOVCD-PCLDesktophp, New folder (2)Abdu codelTM_TEST_10m x Create_TM_10m x bettlowdrm x +       Marker A     Status - C.U.BerskLENOVCD-PCLDesktophp, New folder (2)Abdu codelTM_TEST_10m x Create_TM_10m x bettlowdrm x +       Marker A     Status - C.U.BerskLENOVCD-PCLDesktophp, New folder (2)Abdu codelTM_TEST_10m x Create_TM_10m x bettlowdrm x +       Marker A     Status - C.U.BerskLENOVCD-PCLDesktophp, New folder (2)Abdu codelTM_TEST_10m x Create_TM_10m x bettlowdrm x +       Marker A     Status - C.U.BerskLENOVCD-PCLDesktophp, New folder (2)Abdu codelTM_TEST_10m x Create_TM_10m x bettlowdrm x +       Marker A     Status - Create_TM_10m x Create_TM_10m x bettlowdrm x +       Marker A     Status - Create_TM_10m x bettlowdrm x bettlowdrm x +       Marker A     Status - Create_TM_10m x bettlowdrm x bettlowdrm x +       Marker A     Status - Create_TM_10m x bettlowdrm x bettlo	
Name -       MIES: Lom x: Ceste_DALOm X: Ceste_DALOm X: +         Indod       Indox X: Ceste_DALOm X: +         Indox X: Ceste_DALOm X: Ceste_DALOm X: +       +         Indox +       +	•
abdus       Size (Li), ''''''''''''''''''''''''''''''''''''	۲
inproduktase     559     disp('50 Fake finderprint images redict as Fake = ', num2str(fakecount)];       SholAn     550     rile ['Number of Fake images predict as Real = ', num2str(fakecount)];       SholAn     551     rile ['Number of Fake images predict as Real = ', num2str(fakecount)];       SholAn     552     disp('fill);       AbduSwedBylabrizm     554     disp('fill);       Shertefencem     554     rile ('RALF TOTAL ERROR RATE (HTER) = ', num2str(fuecount1/50)];       Shertefencem     555     disp(FRR);       Shertefencem     556     file(TER);       State     560     disp(FRR);       Sect a file to view details     562     s	
a coulds       50 ril= ('Number of Fake images predict as Fake = ', num2str(fakecounti)];         bill       51 ril= ('Number of Fake images predict as Fake = ', num2str(fakecounti)];         bill       51 ril= ('Number of Fake images predict as Fake = ', num2str(fakecounti)];         bill       51 ril= ('Number of Fake images predict as Fake = ', num2str(fakecounti)];         bill       51 ril= ('Number of Fake images predict as Fake = ', num2str(fakecounti)];         bill       51 ril= ('Number of Fake images predict as Fake = ', num2str(fakecounti));         bill       55 rill         bill       56 rill         bill       57 rill         bill       57 rill         bill       58 rill         58 rill       1         betill       1         betill       58 rill         58 rill       1         59 rill       1         59 rill       1         50 rill       1         56 rill       1         56 rill       1         57 rill       1         58 rill       1         56 rill       1 </td <td></td>	
Standard       551 - ff= ('Humber of Fake images predict as Real = ',num2str(truecountl));         Standard       552 - disp(rfl);         Standard       553 - disp(rfl);         Standard       554 - disp(rfl);         Standard       555 - disp(rfl);         Standard       550 - disp(rfl);         Standard       562 - disp(rfl);         Standard       563 - disp(rfl);         Standa	
Solution       S5       disp(rfl);         S5       S5       filep(rfl);         S5       Forder[*ALS GEWINE RATE(FFR) = ',num2str(ruecount1/50)];         S5       Forder[*ALS GEWINE RATE(FFR) = ',num2str((truecount1/50)];         S5       S5         S5       Forder[*RLS FACE RATE(FFR) = ',num2str((truecount1/50)];         S5       S5         S5       S5         S5       S5         S5       disp(FFR);         disp(FFR);         S6       S6         S6       S6 <td></td>	
Addusted/Ibitizm     55     FCRef 'Fill GENUTIRE RATE (FGR) = ', num2str(truecountl/50)];       S55     FCRef 'Fill GENUTIRE RATE (FGR) = ', num2str(truecountl/50)];       Addusted/optimizm     555       Sett     Addusted/optimizm       Sett     File	
Implementation       Implementation         Implementation       Implementa	
Average/Determine       556 - FFRe['FLS_FARE EATE (FFR) = ', num2str([deccount/50)];         Soft Average/Determine       557 - HTERe['HLS_FARE EATE (HTER) = ', num2str(((truecount/50)+(fakecount/50))/2)];         bets_dissification       v         bets_dissification       v         Select a file to view details       v         read       v         Select a file to view details       v         read       v         Mone en       Value         Number of Real images predict as Seal = 45 Number of Real images predict as Fake = 5         Number of Real images predict as Fake = 42	
Select a file to view details         command Window           Command Window         command Window           Command Window         command Window           Number of Real images predict as Real = 45 Wunder of Real images predict as Fake = 5 Wunder of Real images predict as Fake = 42	
Best5_classification.m         v         555 - disp(FR);           table         550 - disp(FRE);           560 - disp(FRE);         560 - disp(FRE);           561 - disp(FRE);         562 - disp(FRE);           562 - disp(FRE);         563 - disp(FRE);           563 - disp(FRE);         563 - disp(FRE);           564 - disp(FRE);         563 - disp(FRE);           565 - disp(FRE);         563 - disp(FRE);           563 - disp(FRE);         563 - disp(FRE);           564 - disp(FRE);         563 - disp(FRE);           565 - disp(FRE);         563 - disp(FRE);           566 - disp(FRE);         563 - disp(FRE);           567 - disp(FRE);         563 - disp(FRE);           568 - disp(FRE);         563 - disp(FRE);           569 - dis	
Kerkspace     Command Window       Kerkspace     Command Window       Number of Real images predict as Real = 45 Number of Real images predict as Fake = 5 Number of Real images predict as Fake = 42	
Select a file to view details Select a file to view details Command Window Command Window Number of Real isages predict as Real = 45 Number of Real isages predict as File = 42	
Select a file to view details Select	
Select a file to view details  Vorkspace  Vo	
Norkspace     Command Window       Name ^     Value       Inc     1       Inc     1       Inc     0 ref       Inc     1       Inc     0 ref       Inc     0 ref       Inc     1	
Norkspace     Command Window       Name ^     Value       Inst.     1	>
Norkspace     Image: Specific task and t	,
Amme A     Value       Number of Real images predict as Real = 45       Number of Real images predict as Fake = 5       Number of Fake images predict as Fake = 42	
and 1 Author of Real Images predict as Fake = 5 Number of Fake Images predict as Fake = 42	
Number of Fake images predict as Fake = 42	
Avg_T_MD         Outdot         Number of Fake images predict as Real = 8           Avg_F_MD         0.2198         FALS GENUTR #ATE/FGR = 0.16	
Avg_F_NAE 0.0261 Fals Fake Rate(FER) = 0.1	
Avg_F_PSNR 79.5718 HALF TOTAL ERROR RATE (HTER) = 0.13	
Arg.F.SRAMD 0.1901 fr. >> fr.	
avgsmtvvv <	

Figure 38: LDA training result of 100 fingerprint images using 15 IQMs.

## **2. QDA**

📣 MATLAB R2016a											- 6	o x
HOME F	PLOTS APPS	EDITOR	R PU	IBLISH VIEW					614	🖻 🔁 🕐 Search	Documentation	∡ م
Get More Apps App App	age Curve Fitting	Optimization	PID Tuner	System Signal Analysis	Image Instrument Acquisition Control	SimBiology	C MATLAB Coder	Application Compiler	Classification Learner	Distribution Fitting		
FILE					APPS							
🔶 🛋 🖸 📥 +	C: • Users • LENO	VO-PC   Desktop										<del>,</del> 7
Current Folder			🐨 📝 Ed	litor - C:\Users\LENOVO-PC\I	Desktop\4.8.2019 thesis\tm_1	5_classifying.m						⊙ ×
📄 Name 🔺			1 t	tm_25_classifiying_50img.	m 🛛 Main.m 🖂 TES	T_15_(1000).m	X TM_Create_25	i.m ≍ TM_	Create_15.m	tm_15_classifying	g.m × +	_
finger database     f	0_Ida_classifier.txt cem m im ideError.m ror.m deError.m rer.m		<ul> <li>61</li> <li>62</li> <li>63</li> <li>64</li> <li>65</li> <li>66</li> <li>67</li> <li>68</li> <li>69</li> <li>70</li> <li>71</li> <li>72</li> <li>73</li> <li>74</li> <li>75</li> </ul>	<pre>- disp('2-spectra disp(SME); * *spectral phas * SPE = Spectral disp('3-spectral disp('3-spectral disp(SPE); * *gradient magni GME = Gradient disp('4-gradiet</pre>	PhaseError (origImg, al phase error = ') itude error MagnitudeError (orig at magnitude error	distImg); ; Img, distIm						
Workspace												>
Name A AD Ans Avg_F_GME Avg_F_GPE Avg_F_MSE Avg_F_NAE Avg_F_NAE Avg_F_NAE	Value 100x1 double 1 8.5543e-04 7.2278e-04 0.4468 0.2198 7.9124e-04 0.0261 0.9910		New No No No No No No No No No No No No No	mand Window to MATLAB? See resources fo umber of Real images umber of Fake images umber of Fake images ALS GENUINE RATE (FGF ALS FAKE RATE (FRF) = ALF TOTAL ERROR RATE	<pre>s predict as Real = s predict as Fake = predict as Fake = s predict as Real = t) = 0 = 0.02</pre>	= 1 = 50						(c) >
Ava F PSNR	79.5718		v <									>
									script		Ln 652	Col 14

Figure 40: QDA training result of 100 fingerprint images using 15 IQMs.

## 3. Linear SVM

MATLAB R2016a		– 01 ×
HOME PLOTS APPS EDITOR	PUBLISH VEW	🔏 🚊 🕤 🔄 🖨 🕐 Search Documentation 🛛 🔎
	Image         Image <th< th=""><th>Classification Distribution Learner Fiting</th></th<>	Classification Distribution Learner Fiting
FILE	APPS	
= 🕪 🛅 🛜 📙 🕨 C: 🕨 Users 🕨 LENOVO-PC 🕨 Desktop 🕨		•
urrent Folder	Editor - C:\Users\LENOVO-PC\Desktop\4.8.2019 thesis\tm_15_classifying.m	•
Thing of database Thing of the database T	<pre>1 tm_22_destripting_SOmgam X   Mainm X TES_15_(1000),m X TM_Create_25.m X TM 55 Whan Square Error 57 disp(1-Hean SquareError(origImg, distIng); 57 disp(1-Hean SquareError='); 58 disp(1-Hean SquareError='); 59 disp(1-Hean SquareError='); 60 vspectral magnitude error 61 - SKE - spectralMagnitudeError(sigImg, distIng); 62 disp(SRE); 63 disp(SRE); 64 vspectral phase error 64 sspectralPhaseError(origImg, distIng); 65 disp(SSE); 66 disp(SSE); 67 disp(SSE); 67 disp(SSE); 68 disp(SSE); 69 disp(SSE); 69 disp(SSE); 60 disp(SSE); 60 disp(SSE); 61 disp(SSE); 61 disp(SSE); 62 disp(SSE); 63 disp(SSE); 64 disp(SSE); 65 disp(SSE); 66 disp(SSE); 66 disp(SSE); 66 disp(SSE); 67 disp(SSE); 66 disp(SSE); 67 disp(SSE); 67 disp(SSE); 68 disp(SSE); 69 disp(SSE); 60 disp(SSE</pre>	Create_15m X tm_15_classifying m X +
Vorkspace		>
Aame A Value AD 100x1 double ans 1 Avg.F,0D 5558e-04 Avg.F,0E 7.2278e-04 Avg.F,0F 0.4468 Avg.F,ND 0.2198 Avg.F,NAE 00261 Avg.F,NAE 00261 Avg.F,NAE 00261	New to MAILAB? Secretources for Gitting Statted.         Number of Real images predict as Real = 50         Number of Real images predict as Pake = 0         Number of Fake images predict as Real = 0         FALS FAKE RATE (FFR) = 0         FALS FAKE RATE (FFR) = 0	
Ava F PSNR 79.5718	v <	script Ln 662 Col 1

Figure 41: Linear SVM training result of 100 fingerprint images using 15 IQMs.

## 4. Quadratic SVM

Apps       App	📣 MATLAB R2016a							– 🛛 ×
Data base base page       Curve Hing Optimization       PD Tumer       System Supplications       Serie for the seri for the serie for the serie for the serie	HOME PLOTS APP	S EDITOR	PUBLISH VIEW				🕤 定 📑 🕐 Search Docum	entation 🔎 🔺
Image: Solution of the set of the s	Get More Install Package Curve Fitting		ner System Signal Analysis	Image Instrument	<b>2</b>	er Application Classificat	ion Distribution	
Aument Folder       Imped database       Imped				APPS				
Name -	🔶 🔁 🔀 📙 🕨 C: 🕨 Users 🕨 LENO	DVO-PC + Desktop + 4.8	8.2019 thesis 🕨					<b>-</b> ₽
a)       forget database       256       \$274 addient magnitude error         b)       a)       forget database       256       GEE GradientMagnitudeError(origing, dietImg);         256       disp(rddlp;;       256       disp(rddlp;;       256         261       Wgradient magnitude error       261       gradient magnitude error       261         263       disp(rddlp;;       266       GEE GradientQpinseError(origing, distImg);       263         264       disp(rddlp;;       264       disp(rddlp;;       265         264       disp(rddlp;;       264       disp(rddlp;;       265         264       disp(rdlp;;       264       disp(rdlp;;       265         266       V(i,v)=(NSZ, SNR; SNR; SNR; NN; AD; SC; MD; RAMD; NAE; TED; SPE; SNE; GNE; GEE; TCD];       266         261       disp(r(v)=');       266       disp(r(v)=');         266       disp(r(v)=');       266       disp(r(v)=');         266       disp(r(v)=');       266       disp(r(v)=');         266       disp(r(v)=');       266       disp(r(v)=');         266       disp(r(v)=');       266       disp(r(v)=');         267       Qistanda       Qistanda       Qistanda         Avg_f, AD       5334;	Current Folder	۲	Editor - C:\Users\LENOVO-PC\D	esktop\4.8.2019 thesis\tm_15_c	lassifying.m			⊙ ×
a) fogr database       256       % gradient magnitude error         b)       000       00000       0000       0000<	Name 🔺		+1 tm_25_classifiying_50img.r	n X Main.m X TEST_1	5_(1000).m × TM_Create	25.m × TM_Create_15.n	n ≍ tm_15_classifying.m ≍	
Vorkspace         Volu         New to MAILAB? See resources for <u>Getting Started.</u> X           AD         100x1 double         Number of Getting Started.         X           Avg. F_AD         55345-04         Number of Real images predict as Real = 50         Number of Real images predict as Fake = 0           Avg. F_GME         72278-04         Number of Fake images predict as Real = 50         Number of Fake images predict as Real = 0           Avg. F_MD         02198         FALS         FALSUMER KATE (FFR) = 0         Number of Fake images predict as Real = 0           Avg. F_MA         009910         FALS         FALSUMER KATE (FFR) = 0         Number of Fake images predict as Real = 0           Avg. F_MK         09910         C         FALS         OK         V	Bab     matabbyTools-master     results     Tools-master     results     Tools-master     results     Tools-master     results     Tools-master     results     Tools-master     results     AverageOlfference.m     IonimageCaushiyndee.m     GradiemMagnitudeForom     GradiemMagnitudeForom     GradiemMagnitudeForom     Majney.org.rise_index.m     Majney.org.rise_index.m		257         GME = GradientM           258         disp('4-gradien           255         disp(GME);           261         %gradient phase           262         GPE = Gradientp           263         disp('Segret');           264         disp(GFE);           265         disp(GFE);           266         vector of orig           267         w(r,x)=(MSE; x)           266         disp('1-(r');           267         w(r,x)=(MSE; x)           266         disp('1-(r');           267         w(r,x)=(MSE; x)	<pre>agnitudeError(origImg t magnitude error = error haseError(origImg, d: t phase error = '); inal image features</pre>	); LstImg);	; SPE; SME; GME; GPI	:,TCD];	-
Name ^     Value       AD     100x footbe       Not     100x footbe       Name ^     100x footbe       Name ^     10x footbe </td <td></td> <td></td> <td>Command Window</td> <td></td> <td></td> <td></td> <td></td> <td></td>			Command Window					
AD       100x1 double       Number of Real images predict as Real = 50         Awg.F,AD       5538-04         Awg.F,GNE       5538-04         Awg.F,GNE       72278-04         Number of Real images predict as Fake = 0         Number of Fake images predict as Real = 50         Number of Fake images predict as Fake = 0         Awg.F,GNE       72278-04         Number of Fake images predict as Real = 0         FAMS       FALS SEMUINE RATE(FOR) = 0         Awg.F,MD       02198         Awg.F,MAE       00051         Awg.F,MK       09910         Awg.F,FSR       732718         V       C			New to MATLAB? See resources for	Getting Started.				×
	AD 100x1 double Arg_F_AD 8.5543e-04 Arg_F_GME 7.2278e-04 Arg_F_GME 0.4468 Arg_F_GME 0.4468 Arg_F_MD 0.2198 Arg_F_NAE 7.9124e-04 Arg_F_NAE 0.0268 Arg_F_NAK 0.03910		Number of Real images Number of Fake images Number of Fake images FALS GENUINE RATE(FGR FALS FAKE RATE(FFR) = HALF TOTAL ERROR RATE fx >>	predict as Fake = predict as Fake = predict as Real = ) = 0 0	0 50			0
	H Ava F PSNR 79.5718	•				script		Ln 657 Col 3

Figure 42: Quadratic SVM training result of 100 fingerprint images using 15 IQMs.

## 5. Fine KNN

MATLAB R2016a	, T	PUBLISH VIEW			5		🖻 🖻 🕐 Search Doc	umentation
Image: Constant of the second secon	PID Tuner	System Signal Analysis Identification	Image Instrument Acquisition Control	SimBiology M.	ATLAB Coder Applica Comp	ation Classification	Distribution Fitting	
FILE			APPS					
= 🔶 🔄 🔀 📙 🕨 C: 🕨 Users 🔸 LENOVO-PC 🕨 Desktop								-
urrent Folder	۲	Editor - C:\Users\LENOVO-PC\D	esktop\4.8.2019 thesis\tm_15_cla	issifying.m				C
Name A     finger distabase     ib     mattabby/Tools-master     results     Zsign, In, test. 50  da, classifier.txt     A verageDifference.m     BilininageQuilityIndex.m     GradiemphaseEinor.m     GradiemphaseEinor.m     Dingle.guility.score.m		Up         nois         resultsquar           191         disp(H3E):         disp(H3E):         disp(H3E):           192         disp(H3E):         disp(H3E):         disp(H3E):           194         disp(H3E):         disp(H3E):         disp(H3E):           195         disp('6-signatco         disp('6-signatco           196         disp('6-signatco         disp('7-ginatco           196         type         disp('7-speak Signal         disp('7-speak Signal           201         disp('7-speak Signal         disp('7-speak Signal         disp('7-speak Signal           202         disp('9-speak Signal         disp('7-speak Signal         disp('7-speak Signal	<pre>oiseRatio(origImg, di   to noise ratio = ');</pre>	stImg); stImg); g, distImg)	;	TM_Create_15.m	x tm_15_classifying.m	
otalCornerDifference.m (Function)	_ ^ <b>_</b>	<						>
/orkspace		Command Window						
ame -         Value           AD         1002 double           ans         1           Avg_F_AD         8.5548-04           Avg_F_CMR         7.2278-04           Avg_F_OR         0.4468           Avg_F,MS         7.2278-04           Avg_F,MS         0.2198           Avg_F,MS         0.2194           Avg_F,MS         0.0261           Avg_F,FMK         0.03910           Avg F,FNK         0.93910           Avg F,FNR         79.5718	^	New to MATLAB? See resources for Number of Real images Number of Fake images Number of Fake images FALS GENUINE RATE (FGR, FALS FAKE RATE (FFR) = HALF TOTAL ERROR RATE $f_2$ <	predict as Real = predict as Fake = predict as Fake = predict as Real = ) = 0 0	50 0 50 0				

Figure 43: Fine KNN training result of 100 fingerprint images using 15 IQMs.

## 6. Logistic Regression

📣 MATLAB R2016a				– 0 ×
HOME PLOTS APPS	EDITOR PUBLISH VIEW			🔄 🔄 🔁 🕐 Search Documentation 🛛 🔎
Apps App App	timization PD Tuner System Signal Analys	Acquisition Control	MATLAB Coder Application Compiler Learne	
FILE	PC ► Desktop ► 4.8.2019 thesis ►	APPS		•
				1
Current Folder		C\Desktop\4.8.2019 thesis\tm_15_classifying.m	1	⊙ :
Imme -     ingr database     ingr     ingr     ingr database     ingr database     ingr database	<pre>     1     2     3 % clearvars i     4     5     6 - [for i=1:50;         % read origina     8         s - images ='C:\U     10 - hmpfile=wilr(         11 - n=mumel(hmpfil         12 - idx=i;         13 - im=hmpfiles(i </pre>	<pre>isers\LENOVO-PC\Desktop\hp_New fo fullfile(images,'\*.bmp*')); les);</pre>	truecountl fakecountl	n 🗙 tm_15_classifying.m 🗙 (+)
Workspace				>
Name ▲         Value           AD         100x1 double           ans         1           Avg_F_AD         8.5543e-04	Command Window New to MATLAB? See resources Number of Real imag	es predict as Real = 50		
Avg.F.GME         7.2278e-04           Avg.F.GPE         0.4468           Avg.F.MD         0.2198           Avg.F.MSE         7.9124e-04           Avg.F.NAE         0.0261           Avg.F.NK         0.9910	Number of Fake imag	- 0		
Ava F PSNR 79.5718	v <		script	> Ln 5 Col 1

Figure 44: Logistic Regression training result of 100 fingerprint images using 15 IQMs.

# 2. Table and Screenshot of Classification Results For 1000 Fingerprint Images Using 15 IQMs

1000 fingerpint images with 15 IQMs results using 6 clasifiers.

## 1. LDA

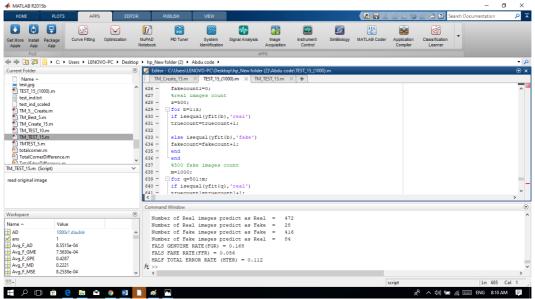


Figure 45: LDA classification result of 1000 fingerprint images using 15 IQMs.

## **2. QDA**

MATLAB R2016a		- 0
HOME PLOTS APPS EDITOR	PUBLISH VEW	🛃 🔚 🔏 😤 💬 🔄 💽 🕐 Search Documentation
Get More Instal Package Apps App App Curve Fitting Optimization P	Image         Image <th< td=""><td>Appleation Classification Distribution Compiler Learner Fitting</td></th<>	Appleation Classification Distribution Compiler Learner Fitting
FILE	APPS	
🔶 🔁 🔁 📙 🕨 C: 🕨 Users 🕨 LENOVO-PC 🕨 Desktop		
Current Folder	Editor - C:\Users\LENOVO-PC\Desktop\hp_New folder (2)\Abdu code\bbbjjhdel.m	
Name -           Name -           Table -           T	<pre>bbjmddm × hbhm ×   + 21 bbjmddm × hbhm ×   + 21 bbjmddm × hbhm ×   + 21 bbjmddm × hbhm ×   + 21 vbfm - PeakSignaltoNoiseEatio 21 vbfm - PiskSignaltoNoiseEatio 21 vbfm - NormalizedCross-Correlation 21 vbfm - NormalizedCross-Correlation (signg, distImg); 22 vbfm vbfm vbfm vbfm vbfm vbfm vbfm vbfm</pre>	
	Command Window	
Name 🔶 Value	New to MATLAB? See resources for <u>Getting Started</u> .	
AD 1000v1 double ans 1 Avg_E_AD & \$3555-04 Avg_E_GME 7,5500-04 Avg_E_GME 0,4287 Avg_F_MD 0,2221 Avg_F_MSE 8,238e-04 Avg_F_MSE 0,0270	Number of Real images predict as Real = 387 Number of Real images predict as Fake = 113 Number of Fake images predict as Real = 0 FALS GENUINE FARE (FRR) = 0 FALS GENUINE FARE (FRR) = 0.226 HALF TOTAL ERROR RATE (HTER) = 0.113 K ≫	
Avg_F_NK 0.9907 Avg_F_SNR 79.2081	v	

Figure 46: QDA classification result of 1000 fingerprint images using 15 IQMs.

#### 3. Linear SVM

Image: State       Image: State <td< th=""><th>📣 MATLAB R2016a</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th>- 0</th><th>×</th></td<>	📣 MATLAB R2016a														- 0	×
Bit is     Curre Withing     Curre Withing <thcurre th="" withing<=""> <thcure th="" withing<="">     Curre Withing&lt;</thcure></thcurre>	HOME	PLOTS APPS	S EDITO	DR	PUBLISH	VEW							@ 🗗 🕐 s	earch Docume	ntation	<b>P</b>
Image: Note:       > Co. 9 Uses:       > LENOVO-PC > Desktop > hp_Newfolder (2) > Abdu code >         Image: Name        Image: Name        Image: Name        Image: Name        Image: Name          Image: Name        I	Get More Install Par	ckage Curve Fitting	$\sim$	_	System	Signal Analysis	Image	Instrument	-	C	Application	Classification	Distribution	-		
arent Folder     O     Editor - CMLess/NENVXO-PCD-Rektophp.Pu.Wew folder (2)/Abdu code/bbbjidd.m     O       Name 4     bbbjidd.m × hbhm × 4       abdu     fnger database     bbbjidd.m × hbhm × 4       202     Wean Square Error       203     Wean Square Error = ');       204     Wist - Managare Error = ');       204     Wist - Managare Error = ');       204     Wist - Managare Error = ');       205     Wist - Managare Error = ');       206     Vist - Stangare Error = ');       207     Vistgrit to noise ratio       208     Vistgrit to noise ratio       209     Vistgrit to noise ratio       200     Vistgrit to noise ratio       201     Vistgrit to noise ratio       202     Vistgrit to noise ratio       203     Vistgrit to noise ratio       204     Vistgrit to noise ratio       205     PSIR = PaskignaltoNoiseRatio(origing, disting);       206     Vistgrit to Noise Ratio       207     Vistgrit to Noise Ratio       208     PSIR       209     Vistgrit to Noise Ratio       201     Vistgrit to Noise Ratio       202     Vistgrit to Noise Ratio       203     Vistgrit to Noise Ratio       204     Vistgrit to Noise Ratio       205     Sitte Signalto								APPS								
Name 4         abdu         fngrd diabase         fngrd diabase         MAMA         abdu         abdu </td <td></td> <td>C: Users ELENC</td> <td>VO-PC   Deskt</td> <td></td> <td>•</td>		C: Users ELENC	VO-PC   Deskt													•
abdu     abdu       Mirst     202       SANAAm     202       Wiest     202       SaNAAm     202       Baduum     202       Baduum     203       Baduum     204       SaNAAm     204       Wistpr('-Hean Square Error       205     Vispr('-Hean Square Error = ');       206     207       207     Vispr('-Hean Square Error = ');       208     Vispr('-Hean Square Error = ');       209     Vispr('-Hean Square Error = ');       206     207       207     Vispr('-Hean Square Error = ');       208     - SIR       209     Vispr('-Error = ');       200     Vispr('-Error = ');       201     Vispr('-Error = ');       203     Vispr('-Error = ');       204     Vispr('-Error = ');       205     Vispr('-Error = ');       206     Vispr('-Error = ');       207     Vispr('-Error = ');       208     Vispr('-Hean Square Error = ');       209     Vispr('-Hean Square Error = ');								w folder (2)\Ab	du code\bbbjjh	idel.m						$\odot$
abdu     001       Minist     001       Min	📄 Name 🔺					m ≍ hhh.m ∷	×[+]									
Index     202     Wean Square Error       3ANAdm     203     WSE Mean Square Error       3ANAdm     204     WSE Mean Square Error       3AbdiswedByEdwinn     205     Wsisp('1-Mean Square Error = ');       3abdum     206     Wsisp('String, distIng);       201     Wsisp(String, distIng);     206       202     Wsisp('String, distIng);     201       203     Wsisp('String, distIng);     203       204     Wsisp('String, distIng);     203       205     Wsisp('String, distIng);     203       206     Wsisp('String, distIng);     203       201     Wsisp('String, distIng);     203       202     Wsisp('String, distIng);     203       203     Msisp('String, distIng);     203       204     Wsisp('Strin				^												^
AbaNA     203 - MSE     203 - MSE     Mask       BANA     BabA     205 + MissQuareError(origing, disting);       BabA     StabA     StapA       BabA     StapA     StapA       Asy Low     StapA     StapA       Asy Low     StapA     StapA       Asy Low     OtapA     StapA		e					P									
abdum abdum abdum abdum abdum abdum abdum abdum abdum abdum abdum abdum abdum abdum abdum abdum abdum abdim abdum abdim abdim abdim bbijdef abdim abdim bbijdef abdim a								rigTmg di	t Tmar) :							
abddsmed/m     205     stirp (MSE);       bddsmed/med/shirtm     205     stirp (MSE);       bddsmed/shirtm     207     stippal.tonoise ratio       bddsmed/med/shirtm     207     stippal.tonoise ratio       bddsmed/shirtm     207     stippal.tonoise ratio       bddsmed/shirtm     208     stippl.tonoise ratio       bbdshirth     209     stippl.tonoise ratio       bbdshirth     209     stippl.tonoise ratio       bbdshirth     209     stippl.tonoise ratio       bbdshirth     209     stippl.tonoise ratio       bbdshirth     201     tippl.tonoise ratio       bbdshirth     201     tippl.tonoise ratio       bbdshirth     201     tippl.tonoise ratio       bbdshirth     201     tippl.tonoise ratio       201     tippl.tonoise     201       202     tippl.tonoise ratio     201       202     tippl.tonoise     201       203     tippl.tonoise ratio     201       204     tippl.tonoise     201       203     tippl.tonoise     201       204     tippl.tonoise     201       204     tippl.tonoise     201       205     tippl.tonoise     201       205     tippl.tonoise     201									(cing),							
206     207       abduum     206       abduum     206       abduum     208       bebindem     208       bebindem     211       212     ¥eagnalto to noise ratio = ');       213     FBRM       ame ^     Value       An     7000ri double       ans     1       Ang_LOM     7050bc-04       Ang_LOM     7560bc-04       Ang_LOM     7560bc-04       Ang_LOM     7560bc-04       Ang_LNM     0.9271       Ang_LNM     0.9271       Ang LNG     0.0270       Ang LNG     0.9271       Ang LNG							Square Fr	101 - 7,								
Implementation     207     \$signal to noise ratio       Implementation     207     \$signal to noise ratio       Implementation     208     Signal to noise ratio       Implementation     208		Tabaia na				oursp(mar),										
And and addition common with the set of the set		rabriz.m				signal to no	ise ratio									
Avergeptifferencem     205     \$11ap;(*=lgnato to noise ratio = ');       bettyl.kom     10     \$11ap;(*=lgnato to noise ratio = ');       bettyl.kom     211     \$11ap;(*=lgnato to noise ratio = ');       bettyl.kom     211     \$11ap;(*=lgnato to noise ratio = ');       bettyl.kom     211     \$11ap;(*=lgnato to noise ratio = ');       211     212     \$1ap;(*=lgnato to noise ratio = ');       212     bettyl.kom     213 =       bettyl.kom     213 =     PSNR = PeakSignal to Noise Ratio (origim;, distimg);       c     c     >       arg.f.A0     1000r facels     Number of Real images predict as Real =     406       Ang.f.OM     7550e-04     Number of Real images predict as Real =     14       Ang.f.OM     0221     Number of Real images predict as Real =     19       Ang.f.M6     02221     FALS FARE FATE(FRR) =     0.038       Ang.f.M6     02270     Ang.f.M6     02070       Ang.f.N6     02070     Ang.f.M6     02070       Ang.f.PNK     09077     0.033     Activate Windows       Activate Windows								o (origImg.	distImg):							
betsjindedm     210     \$119(SBR);       betsjindedm     211     \$119(SBR);       betsjindedm     212     \$Peak Signal to Noise Ratio       betsjindedm     213     PSNR = PeakSignaltoNoiseRatio(origing, disting);       213     SIS     PSNR = PeakSignaltoNoiseRatio(origing, disting);       214     Siste-other     Siste-other       ans     Value     Number of Real images predict as Real = 486       Ang_f.DR     7,550-04     Number of Real images predict as Fake = 481       Ang_f.NE     0,2221     Ang.F.NE     0,0220       Ang_f.NE     0,2221     FALS GENUTIF RATE(FRR) = 0.038       Ang_f.NM     0,9077     FALS GENUTIF RATE(FRR) = 0.038       Ang.F.NK     0,9077     Go to Settings to activate Windows	AverageDiffere	nce.m														
betty, dustification.m     111     112     112     111     111       betty, dustification.m     112     112     112     112     112       betty, dustification.m     113     113     113     113     113       betty, dustification.m     113     113     113     113     113       betty, dustification.m     113     113     113     113     113       ordspace     013     113     113     113     113     113       ordspace     013     113     113     113     113     113       Apg. Lot     1000/ fdouble     110     113     113     113     113       Apg. Lot     1000/ fdouble     110     113     113     113     113       Apg. Lot     75500-04     113     113     113     113     113       Apg. Lot     0.4221     113     113     113     113     113       Apg. Lot     0.4221     113     113     113     113     113       Apg. Lot     0.2221     113     113     113     113     113       Apg. Lot     0.221     113     113     113     113     113       Apg. Lot     0.221     113     113<																
Detroit/ntm     115 =     PSNR = PeakSignaltoNoiseRatio(origimg, disting);       otals     215 =     PSNR = PeakSignaltoNoiseRatio(origimg, disting);       otals     Command Window       ans     1       Ang_f.AD     85356-04       Ang_f.OH     Command Window       Ang_f.OH     75800-04       Ang_f.OH     Oct221       Ang_f.NG     82358-04       Ang_f.MG     02201       Ang_f.MG     02201       Ang_f.NG     02070       Ang_f.NG     02070       Ang_f.PMS     020270       Ang_f.PMS     020270       Ang f.PMS     09977       Ang f.PMS     09977 <th< td=""><td></td><td>ation.m</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></th<>		ation.m														
ATT     VIS     PERKS QHAILONOISERALIO (DELYMM), dist.im()?       table     >       orkspace     C       ans a     Value       AD     1000+ double ang_f.AD       AD     1000+ double ang_f.AD       ASSISE-04       Ang_f.BH     7.5500- 04       Ang_f.ME     5.5150- 04       Ang_f.ME     7.5500- 04       Ang_f.ME     0.2221       Ang_f.ME     0.2221       Ang_f.ME     0.2221       Ang_f.ME     0.2221       Ang_f.ME     0.2221       Ang_f.ME     0.2221       Ang_f.ME     0.2201       Ang_f.ME     0.2021       Ang_f.ME     0.2021					212	Peak Signal	to Noise R	atio								
tabis				~	213 - 1	PSNR = PeakSi	gnaltoNois	eRatio(orio	Img, dist	Img);						
orkspace         Command Window           ame +         Value         New to MATLAST See resources for <u>Getting Started.</u> AD         1000rl double         Number of Started.           Arg_F_AD         85515e-04           Arg_F_GMF         0.4287           Arg_F_MSE         82538e-04           Arg_F_MSE         0.2221           Arg_F_MSE         0.0270           Arg_F_MSE         0.9907           Arg_F_MSE         0.9907           Arg_F_MSE         0.9907           Arg_F_MSE	letails			~	<											>
AD     1000x1 double     AD       AD     1000x1 double     AD       Ang_LAD     85355e-04     Anwaber of Real images predict as Real = 466       Ang_LGME     7580e-04     Number of Real images predict as Fake = 14       Ang_LGME     0.2221     Number of Fake images predict as Real = 19       Ang_LMS     82358e-04     Anwaber of Fake images predict as Real = 19       Ang_LMS     82358e-04     Anwaber of Fake images predict as Real = 19       Ang_LNME     0.0221     FALS GRUINE RATE (FGR) = 0.038       Ang_LNME     0.0270     Ang LAN       Ang LF PSME     0.9907     .       Ang LF PSME     72,081     .					Command Win	dow										
AD         1000-1000ble         AD           srs         1         Number of Real images predict as Real = 486           Ang_F,AD         8.555-04           Ang_F,GM         8.555-04           Ang_F,GM         75600-04           Ang_F,GM         0.221           Ang_F,MS         0.221           Ang_F,MS         0.221           Ang_F,MS         0.220           Ang_F,MS         0.0270           Ang_F,MS         0.0270           Ang_F,MS         0.9077           Ang_F,MS         0.9077           Ang F,MS         0.9077           Ang F,MS         0.9077	Vame +	Value			New to MATLA	AB? See resources fo	or <u>Getting Starte</u>	ed.								
ans     1     Number of Neal images predict as Neal = 190       Ang.f.AD     S555_04     Number of Real images predict as Neal = 190       Ang.f.OM     7550_04     Number of Real images predict as Neal = 14       Ang.f.OM     7550_04     Number of Real images predict as Neal = 14       Ang.f.OM     7550_04     Number of Real images predict as Neal = 19       Ang.f.MS     0.2271     Number of Real images predict as Real = 19       Ang.f.MS     82538_04     Ang.F.NE       Ang.f.NM     0.2070     FALS GENUIUE RATE (FGR) = 0.033       Ang.f.NM     0.907     Activate Windows       Ang.f.SN     0.907     V       Kar FSNR     72,081     V				_												
Avg.F.A0         8.555-04         Number of Real images predict as Fake = 14           Avg.F.060         7.5600-04         Number of Real images predict as Fake = 40.           Avg.F.061         0.2201         Number of Rake images predict as Real = 19           Avg.F.062         0.2201         FALS GRUINE RATE (FGR) = 0.038           Avg.F.N62         0.2202         FALS FAKE RATE (FFR) = 0.028           Avg.F.N6         0.0270         HALT TOTAL ERROR RATE (HER) = 0.033           Avg.F.N6         0.9077         K           Avg.F.D8         0.9077         K				<u> </u>												
Avg.F.0ME         7,5500-04         Number of rake images predict as rake = 491           Avg.F.0ME         0.4287         Number of rake images predict as rake = 491           Avg.F.0ME         0.4287         Number of rake images predict as rake = 19           Avg.F.0ME         0.2201         FALS GENUINE RATE (FGR) = 0.038           Avg.F.NK         0.0270         HALF TOTAL ERROR RATE (HTER) = 0.033           Avg.F.NK         0.9907            Avg.F.DNK         0.9907		8.5515e-04														
Avg_F_00         0.4287         Number of Fake Images predict as Real = 19           Avg_F_M0         0.2221         FALS GRUINE RATE (FGR) = 0.033           Avg_F_M8         8.2538-04         FALS FAKE RATE (FFR) = 0.028           Avg_F_M8         0.0270         HL FORL FROM RATE (HTER) = 0.033           Avg_F_M8         0.9907         K           Avg_F_M8         0.9907         K																
Mug_r,Mod         Guzzi         Falls	Avg_F_GPE	0.4287						as Real =	19							
Mug_rNac allose of Aug_rNac all	Avg_F_MD	0.2221														
Activate Vintovis Ava F PSNR 792081 V C Go to Settings to activate Windows.	Avg_F_MSE															
Ava F PSNR 79,2081 V Go to Settings to activate Windows.	Avg_F_NAE					AL ERROR RATI	E (HTER) =	0.033					Activate V	Vindows		
Ava F PSNR 79.2081											_			as to activate	Windows.	
- script Ln 207 Col	Ava F PSNR	79.2081		*	<											

Figure 47: Linear SVM classification result of 1000 fingerprint images using 15 IQMs.

## 4. Quadratic SVM

HOME PLOTS APPS EDITO	R PUBLISH VIEW	🚡 💼 😒 🕞 🔁 🕐 Search Documentation 🛛 🔎
Image: Second system         Image: Se	PD Tuner System Signal Analysis Image Instrument SimBiology MATLAS Coder Application Clu	issification Distribution + Learner Fitting
FILE	APPS	
🖶 🔶 🔁 🔁 📙 🕨 C: 🕨 Users 🕨 LENOVO-PC 🕨 Deskto		•
Current Folder	Zéittor - CaUsers/LENOVO-PC/Desktop/hp_New folder (2)/Abdu code/bbbjjhdel.m     E bbbjjhdel.m X hhh.m X +	$\odot$
B abdu B abdu B finger database B abdum B abdum AbduSavedByTabriz.m B abdum AbduSavedByTabriz.m B abdum B abdum	<pre>is; - uisp('fird;' biginal image to biscored image should be of same</pre>	
	Commend Wendow	
Norkspace	Command window     New to MATLAB? See resources for Getting Started.	
Name ▲ Value AD 1000/1 double ✓ ann 1 Arug F_AD 8,55156-04 → arug F_GME 7,5608-04 → arug F_GME 0,4287 → arug F_MS 8,2538-04 → arug F_MS 8,2538-04 → arug F_NK 0,9907 → arug F_NK 0,9907 → arug F_NK 0,9907	Number of Real images predict as Real = 477 Number of Real images predict as Fake = 23 Number of Fake images predict as Fake = 470 Number of Fake images predict as Real = 30 FALS FAKE EATE(FER) = 0.06 FALS FAKE EATE(FER) = 0.06 HALF TOTAL ERROR RATE (HTER) = 0.053 A >>	Activate Windows Go to Settings to activate Windows

Figure 48: Quadratic SVM classification result of 1000 fingerprint images using 15 IQMs.

## 5. Fine KNN

HOME PLOTS	APPS	EDITOR	PUBLISH	VEW						. <u>4 É 2</u>	600	Search Documentation	م
	e Fitting Optimize		er System Identification	Signal Analysis	Image Acquisition	Instrument Control	SimBiology	MATLAB Coder	Application Compiler	Classification Learner	Distribution Fitting	•	
	► LENOVO-PC ►	Desktop + hr	New folder (2) Ab	du code 🕨	· · · · · ·	rrə							•
urrent Folder			Z Editor - C:\Users		esktop\hp Nev	/ folder (2)\Abd	u code\bbbijho	del.m					$\odot$
Name - shdu finger database hiffest 3 JaNDAm 3 Jabdum 3 Jabdum 4 AdulSaved97Jabrizm 3 Adulaum 3 shdukam 4 AdulSaved97Jabrizm 3 shdukam 4 AdulSaved97Jabrizm 4 AdulS			62 - MSE 63 %dis 64 %dis 65 66 %spe 67 - SME 68 %dis	<pre>Main Square E Main Main Main Main Main Main Main Main</pre>	rror reError(or Square Err itude erro MagnitudeE	or = '); r rror(origIn	ng, distIm	(g);					
tails		^	Command Window										
orkspace		۲			a								_
Imme         Value           AD         1000x1 dout           ans         1           Avg_F_GD         8.5515e-04           Avg_F_GME         7.5500e-04           Avg_F_GME         0.4287           Avg_F_GME         0.4287           Avg_F_MAE         0.4287           Avg_F_NAE         0.0270           Avg_F_NAE         0.0270           Avg_F_FNAE         0.0270           Avg_F_FNAE         9.907           Avg_F F2NR         79.2081	le		New to MATLAR? S Number of F Number of F Number of F FALS FAKE F HALF TOTAL $f_{\nabla} >> <$	eal images eal images ake images ake images E RATE (FGR ATE (FFR) =	predict a predict a predict a predict a ) = 0.102 0.032	s Real = s Fake = s Fake = s Real =	484 16 449 51				Activate ' Go to Settin	Windows gs-to-activate Windo	lows.
										script		Ln 68	89 Col
-										script		4:47	

Figure 49: Fine KNN classification result of 1000 fingerprint images using 15 IQMs.

## 6. Logistic Regression

📣 MATLAB R2016a														-	o >
номе	PLOTS	APPS	EDITOR	۱	PUBLISH	VIEW					88.	419	🖻 🔁 🕐 Sea	rch Documentation	م ا
	ackage Curve I		mization	PID Tuner	System Identification	Signal Analysis	Image Acquisition	Instrument Control	SimBiology	MATLAB Coder	Application Compiler	Classification Learner	Distribution Fitting	•	
FILE								APPS							
🕈 🌩 🖪 🔁 📒	► C: ► Users ►	LENOVO-PC	<ul> <li>Desktop</li> </ul>	► hp_N	lew folder (2) 🕨	Abdu code 🕨									-
Current Folder				•	🜌 Editor - C:\Us	ers\LENOVO-PC\	Desktop\hp_N	ew folder (2)\Ab	du code\bbbjjh	idel.m					$\odot$
🗋 Name 🔺					bbbjjhdel.m	🛛 🗶 📄 hhh.m	≤]+]								
test.ipg						- NOTMATIZE				cimd):					
test.jpg TEST 15 (100						sp('8-Norma	lized Cros	s-Correlati	on = ');						· · · ·
test ind.txt	0).m					sp (NK) ;									
test ind scale	ad				221										
testhlfiiii.m						erage Diffe									
TM_5_Create	e.m					<ul> <li>AverageDi</li> </ul>			stImg);						
M Best 5.m						sp('9-Avera	ge Differe	nce = ');							
M Create 1						.sp (AD) ;									
TM TEST 10.					226										
TM_TEST_15.	m					ructural Co									
TMTEST_5.m						= Structura									
totalcorner.m						sp('10-Stru	ctural Con	tent = ');							
🔊 TotalCornerD	lifference.m					sp(SC);									
🛛 🙆 TotalEdgeDiff	ference.m				231										~
TotalEdgeDifference	s.m (Function)			~	<										>
Workspace					Command Wind										
Name 🔶	Value				New to MATLAE	? See resources fo	er <u>Getting Start</u>	ed.							
Avg R RAMD	0.2295														
Avg R SC	1.0158			<u>^ </u>		Real image			466						
Avg_R_SME	199.7606					Real image			34						
Avg_R_SNR	28.3501				Number of	Fake image	s predict	as Fake =	481						
Avg_R_SPE	0.0023				Number of	Fake image	s predict	as Real =	19						
Avg_R_TCD	0.0233				FALS GENU	INE RATE (FG	R) = 0.038								
Avg_R_TED	0.0077				FALS FAKE	RATE (FFR)	- 0.068								
b	500					L ERROR RAT	E (HTER) =	0.053							
best15	15x1 double				fx, >>										
🗄 bmpfiles	512x1 struct			~	<										)
												script		Ln 22	26 Col 1

Figure 50: Logistic Regression classification of 1000 fingerprint images using 15 IQMs.

# 3. Table and Screenshot of Training Results For 100 Fingerprint Images Using 10 IQMs

100 fingerprint images with 10 IQMs result using 6 classifiers.

## 1. LDA

HOME PLOTS APPS EDITO	R PUBLSH VEW Search Documentation	م
Image: Note of the state of the st	Image         Image <th< th=""><th></th></th<>	
FILE	APPS	
Name 🔺	<pre>Z fator: CiUseniLENOVO-PC.Dexhop/hp.Newfolds (2)Abdu code/TM_TEST_ID.m  Z fator: CiUseniLENOVO-PC</pre>	C
Select a file to view details	562 b 583 ¢c4	>
onteparce C ontepa	Command Window Number of Real images predict as Real = 45 Number of Fake images predict as Fake = 5 Number of Fake images predict as Real = 11 FALS GENUINE RATE(FGR) = 0.22 FALS FAKE RATE(FFR) = 0.12 // ALS FAKE RATE(FFR) = 0.16 	

Figure 51: LDA training result of 100 fingerprint images using 10 IQMs.

## 2. QDA

MATLAB R2016a		- 🗗 🗙
HOME PLOTS APPS EDT	PO Tuner System Signal Analysis Image Instrument SimBiology MATLAB Coder Application	on Classification Distribution
FILE	APPS	
	op > hp_New folder (2) > Abdu code >	•
Current Folder		•
MatcutualContent.m           StructualSindiniyinder.m           StructualSindiniyinder.m           Istesting           TST_15_1(0000.m           Istesting           Istesting           Matcut           Istesting           Matcut           Istesting           Matcut           Istesting           Matcut           Istesting           Matcut           Imatcut           Imatcut	<pre>&gt; \$73 - disp(zf); \$74 - disp(zf); \$75 - disp(zf); \$76 - disp(zf); \$77 - disp(FR); \$77 - disp(FR); \$79 - disp(HTER); \$79 - disp(HTER); \$59 - 501 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50</pre>	
M_TEST_10.m (Script)		,
Norkspace Value	New to MATLAB? See resources for <u>Getting Started</u> .	
ans         1           Avg_F_OPE         0.4468           Avg_F_IMD         0.2198           Avg_F_IAE         0.0261           Avg_F_PSNR         79.5718           Avg_F_RATD         0.1552	<ul> <li>Number of Real images predict as Real = 48</li> <li>Number of Real images predict as Fake = 2</li> <li>Number of Fake images predict as Fake = 50</li> <li>Number of Fake images predict as Real = 0</li> <li>FALS GENUTUR RATE(FGR) = 0</li> </ul>	
Avg_F_SME 0.0045 Avg_F_SNR 28.5125 Avg_F_SPE 142.4203	FALS FARE RATE (FFR) = 0.04 HALF TOTAL ERROR RATE (HTER) = 0.02 $ \vec{F}_{\rm c}>>$	Activate Windows Go to Settings to activate Windows.
Ava F TCD 0.0344	v (	so to settings to activate windows.

Figure 52: QDA training result of 100 fingerprint images using 10 IQMs.

📣 MATLAB R2016a		– Ø ×
HOME PLOTS APPS EDITOR	PUBLISH VEW	, 🔚 🔏 🖄 😂 😂 🔁 🕐 Search Documentation 🛛 🔎
Get More Instal Package Curve Fitting Optimization P	Tuner System Signal Analysis Image Instrument SimBiology MATLAB Coder Appli	ication Classification Distribution Fitting
FILE	APPS	
🗏 🔶 🔁 📒 🕨 C: 🕨 Users 🕨 LENOVO-PC 🕨 Desktop		•
Current Folder	Editor - C:\Users\LENOVO-PC\Desktop\hp_New folder (2)\Abdu code\TM_TEST_10.m	
🗋 Name 🔺	bbbjjhdel.m × hhh.m × Create_TM_10.m × Class_best_10.m × TM_TEST_10.m >	< unique.m × +
	<pre>A 126 Whormalized Absolute Error 127 NRE = NormalizedAbsoluteError(origImg, distImg); 128 tdisp('13-NormalizedAbsoluteError = '); 129 tdisp('13-NormalizedAbsoluteError = '); 130 131 WTotalEdgeDifference 132 - TED = TotalEdgeDifference(origImg, distImg); 133 tdisp('14-TotalEdge Difference = '); 134 tdisp(TED); 135 136 WTotalCornerDifference 137 - TCD = TotalCornerDifference(origImg, distImg); 13 tdisp('15-TotalCornerDifference = ');</pre>	
	Commond MGs down	
Workspace	New to MATLAB? See resources for Getting Started.	
Name +         Value           ans         1           Avg.E. (5FE         0.4468           Avg.F. MO         0.2198           Avg.F. F.NE         0.0218           Avg.F. PARM         0.0512           Avg.F. F.NE         0.0045           Avg.F. SNR         0.0045           Avg.F. SNR         26.5125           Avg.F. SNR         142.403	There to markeds bereadures to <u>Manual Annab</u> Mumber of Real images predict as Fake = 0 Number of Fake images predict as Fake = 50 Number of Fake images predict as Fake = 50 Number of Fake images predict as Real = 0 FALS GENUTUR EATE(FGR) = 0 HALS TOTAL ERROR RATE (HTER) = 0 $K_2 > 5$	Activate Windows Go to Settings to activate Windows

Figure 53: Linear SVM training result of 100 fingerprint images using 10 IQMs.

# 4. Quadratic SVM

Image: Serie Compare       Image: Serie Compare <td< th=""><th>- 0</th></td<>	- 0
Image: Compare       Image	🗟 📄 😨 Search Documentation
<pre>     C &gt; Uses &gt; ELNOVO-PC &gt; Desitop &gt; hp.New folder (2) &gt; Abdu code &gt;     Intent Folder</pre>	
Immet Folder       Immet Folder         Immet Ander       Immet Addution         abdut       Immet Addution         hiftest       Immet Addution         3ANDALm       Immet Addution         abdut       Immet Addution         Badut       Immet Addution	
Name =       Abdu       Name =       Ibbjiddim × Nhhm × CorateTM.(Dm ×)       Class_bet.(Dm ×)       Mm_TEST_10.m × unique.m × +         Abdu       66       Varpeotral magnitude error       68       Varpeotral magnitude error       69         Abdum       88 WE SpectralNagnitudeToro(origing, distIng);       70       Varpeotral phase error       71         Abdum       70       Varpeotral phase error       71       Varpeotral phase error       71         Abdum       74       SPE = SpectralPhaseLtrow(origing, distIng);       75       Varpeotral phase error       71         Mathem       74       SPE = SpectralPhaseLtrow(origing, distIng);       75       Varpeotral phase error       71         Mathem       78       Varpeotral phase error       71       71       71         Mathem       78       Varpeotral phase error       71       71       72         Mathem       78       Varpeotral phase error       71       71       71       71         Mathem       78       Spectral phase       72       72       72       73       74         Mathem       78       Spectral phase       74       76       76       76       77         78       Spectral phase       76       Specon	
Orkspace         O           anse         Value           anse         1           Ang.F.OF         0.4468           Ang.F.MD         0.2198           Ang.F.PAR         0.0261           Ang.F.PAR         0.0261           Ang.F.PAR         0.0501           Ang.F.SME         0.0455           Ang.F.SME         0.0454           Ang.F.SME         0.0454           Ang.F.SME         0.0454           Ang.F.SME         0.0454           Ang.F.SME         0.0454	*
ame A         Value           ame A         1           Avg.FoF         0.4683           Avg.FAD         0.2188           Avg.FAD         0.2188           Avg.FAD         0.2051           Avg.FAD         0.2051           Avg.FAD         0.0251           Avg.FANA         0.0251           Avg.FANA         0.152           FALS FEAR         FALS GEWUINE RATE (FGR) = 0           Avg.FSNM         283125	
Avg_F_SPE 142.4203 / Avg F TCD 0.0344 / <	Activate Windows Go to Settings to activate Windows.

Figure 54: Quadratic SVM training result of 100 fingerprint images using 10 IQMs.

📣 MATLAB R2016a		– o ×
HOME PLOTS APPS EDITOR	PUBLISH VIEW	🞖 🔚 🚣 🔓 😒 🗟 🛱 🕐 Search Documentation 🛛 🔎 🕇
Get More Install Package Curve Fitting Optimization P	) Tuner System Signal Analysis Image Instrument SimBiology MATLAB Coder App	pication Classification Distribution ompiler Learner Fitting
FILE	APPS	and the second se
⊨ 🔶 🛅 🎏 📙 🕨 C: 🕨 Users 🕨 LENOVO-PC 🕨 Desktop		-
Current Folder	Editor - C:\Users\LENOVO-PC\Desktop\hp_New folder (2)\Abdu code\Class_best_10.m	⊙ :
Name 🔺	bbbjjhdel.m × hhh.m × Create_TM_10.m × Class_best_10.m × TM_TEST_10.m	× unique.m × +
a) abdu b) abdu b) abdu b) abdu b) abdu b) abdum b) abdu	<pre>A 81 kdisp(GKE); 82 83 kgradient phase error 84 - GFE = GradientphaseError(origImg, distImg); 85 kdisp('5-gradient phase error = '); 86 kdisp(GFE); 87 88 89 ksignal to noise ratio 90 - SNR = SignaltoNoiseEatio(origImg, distImg); 91 kdisp('6-signato to noise ratio = '); 92 kdisp(SNR); 93 sdisp('S-signato to noise ratio = '); 94 kdisp(SNR); 94 kdisp('SNR); 95 kdisp('SNR); 95 kdisp('SNR); 96 kdisp('SNR); 96 kdisp('SNR); 97 kdisp('SNR); 96 kdisp('SNR); 96 kdisp('SNR); 97 kdisp('SNR); 96 kdisp('SNR); 97 kdisp('SNR); 97 kdisp('SNR); 98 kdisp('SNR); 98 kdisp('SNR); 99 kdisp('SNR); 99 kdisp('SNR); 90 k</pre>	, v
	Command Window	
Workspace	Command window     New to MATLAB? See resources for <u>Getting Started</u> .	
Name →         Value           an         1           Avg. F, GFE         0.427           Avg. F, MO         0.2221           Avg. F, MO         0.2221           Avg. F, MA         0.0270           Avg. F, MA         0.0202           Avg. F, SAR         0.0029           Avg. F, SAR         0.0009           Avg. F, SAR         2.1201           Avg. F, SAR         1.83251	<pre>500 real fingerprint images result : Number of Real images predict as Real = 483 Number of Real images predict as Fake = 17 500 Fake fingerprint images result : Number of Fake images predict as Real = 460 Number of Fake images predict as Real = 40 FALSE GENUINE RATE (FGR) = 0.08 FALSE TARK ERITE(FGR) = 0.034 /f HALF TOTAL ERROR RATE (HTER) = 0.057 </pre>	Activate Windows Go to Settings to activate Windows.
Ava F TCD 0.0280		script In 1 Col 1

Figure 55: Fine KNN training result of 100 fingerprint images using 10 IQMs.

HOME PLOTS APPS EDITOR	PUBLISH VIEW	🖲 📄 🔁 🕐 Search Documentation 🛛 🔎
More Install Package Curve Fitting Optimization PD Tur	er System Sgnal Analysis Image Instrument SmBology MATLAB Coder Application Cassification	n Distribution Fitting
FILE	APPS	
🔶 💽 💭 📙 🕨 C: 🕨 Users 🕨 LENOVO-PC 🕨 Desktop 🕨 hp		
Name	Editor - C\Users\LENOVO-PC\Desktop\hp_New folder (2)\Abdu code\TM_TEST_10.m     bbbjihdel.m × hhh.m × Create TM 10.m × Class best 10.m × TM TEST 10.m × unique.m × ·	+
abu A	<pre>vose = 'nearoquarterror(virging, 'uirving,' 204 % disp('l-Hean Square Error = '); 205 % disp(NSE); 206 207 % signal to noise ratio 208 = SNR = SignalroNoiseRatio(orging, distImg); 209 % disp('d-signato to noise ratio = '); 210 % disp('SNR); 211 % Peak Signal to Noise Ratio = '); 213 = PSNR = PeakSignaltoNoiseRatio(origing, distImg); 214 % disp('7-Peak Signal to Noise Ratio = '); 215 % disp(PSNR); 216 &lt;</pre>	
ails	Command Window	`
rkspace 💿	New to MATLAB? See resources for Getting Started.	
me ▲ Value  ins:  1  vag.F.26F  0.4468  vag.F.MD  0.2198 vag.F.24A  0.0261  vag.F.25NR  0.0261  vag.F.25NR  0.0392  vag.F.25NR  0.0395  vag.F.25N  0.0345  vag.F.25P  142.4203  vag.F.25P  142.420  143.42  1	Number of Real images predict as Real = 50 Number of Real images predict as Fake = 0 Number of Fake images predict as Fake = 50 Number of Fake images predict as Real = 0 Fals GEUUNE RATE (GPS) = 0 Fals FAKE RATE (FTR) = 0 Half TOTAL ERROR RATE (HTER) = 0 $f_2$ >	Activate Windows Go to Settings to activate Windows-

Figure 56: Logistic Regression training result of 100 fingerprint images using 10 IQMs.

# 4. Table and Screenshot of Classification Results For 1000 Fingerprint Images Using 10 IQMs

1000 Fingerprint images with 10 IQMs result using 6 classifiers:

## 1. LDA

HOME PLOTS APPS E	TOR PUBLISH VIEW	🖥 🔏 🚊 🕤 😅 🖨 🕐 Search Documentation 💦 🔎
t More Instal Package App FILE	Marko         PD Tuner         Stystem         Signal Analysis         Image Image         Image         I	Coder Application Classification Compiler Learner
	top + hp_New folder (2) + Abdu code +	-
I Name A I testing A I testin	O         E Stor- C.VLenkIENVO-PCD Exktphk, New folder (∂/Abda cod+TST_15,1000).m           1         TM_Creat.IST         XTTST_15,0000.m         XTTST_15,0000.m           224         fakecount=10;         XTTST_15,0000.m         XTTST_15.m         X         +           626         fakecount=10;         XTTST_15.m         X         +         +           626         fakecount=10;         XTTST_15.m         X         +           627         freat.Images count         +         +         +           628         = Stor.b=1:z;         +         +         +         +           630         = fisecount=truecount=1;         +         +         +         +         +           631         truecount=fakecount=1;         +	e
orkspace	Command Window           O         Number of Real images predict as Real = 469	
me         Value           AD         1000x1 double           avg_F_AD         8.5515=04           Avag_F_GME         7.5530=04           Avag_F_GPE         0.4287           Avag_F_MD         0.2221           Avag_F_MSE         8.2538=04	Number of Real images predict as Fake = 31           Sup Fake fingerprint images result :           Number of Fake images predict as Fake = 414           Muster of Fake images predict	
		script Ln 685 Col 1

Figure 57: LDA Classification result of 1000 fingerprint images using 10 IQMs.

HOME PLOTS APPS EDITO	R	PUBLISH VIEW				68	4 <b>b c</b> 5	0 2 5 0	Search Documentation	Q
et More Install Package Apps App FILE	PID Tuner	r System Signal Analysis Identification	Image Inst	ument SimBiology	MATLAB Coder	Application Compiler	Classification Learner	Distribution Fitting	·	
🖙 🔁 🔀 📙 🕨 C: 🕨 Users 🕨 LENOVO-PC 🕨 Deskto	p ► hp_l	New folder (2) 🕨 Abdu code 🕨								-
urrent Folder	۲	Editor - C:\Users\LENOVO-PC\	Desktop\hp_New folde	r (2)\Abdu code\Class_b	pest_10.m				1	O
Name 🔺		bbbjjhdel.m 🛛 hhh.m 🗄	Create_TM_10.m	× Class_best_10.m	× +					
abdu finger datases hiffrest 3AND4 m 3bdu4 m 3bdu4 m Abdu5xed8yTabriz m 3 Abdu5xed8yTabriz m 3 Abdu4 m 4 Averag0fifterioc m bets2,classification m bets2,classifica		128         \$disp('13-Norm           129         \$disp('NAE);           130         \$Total Edge Di           131         \$Total Edge Di           132         TED = TotalEdg           133         \$disp('14-Tota           134         \$disp(TED);           135         \$TotalCornerDi           136         \$TotalCornerDi           137         TCD = TotalCornerDi           138         \$disp('15-Total           138         \$disp('CD);	alized Absolute fference eDifference(ori 1 Edge Differen fference	gImg, distImg); ce = '); rigImg, distImg)	-					,
tails	~	<								>
/orkspace		Command Window								
Imme         Value           ans         1           Ang-f_OPE         0.4287           Ang-f_IMD         0.2211           Ang-f_IMD         0.2211           Ang-f_FMD         0.2211           Ang-f_FMD         0.2211           Ang-f_FMD         0.2201           Ang-f_FMD         0.2201           Ang-f_FSMR         0.2002           Ang-f_FSMR         0.0040           Ang-f_SME         0.0040	^	Newto MATLAB? Secresources fr Number of Real image: Number of Real image; 500 Fake fingerprint Number of Fake image; FaLSE GENUINE RATE (FFR) HALF TOTAL ERROR RATI É/ > <	s predict as Re. s predict as Fa images result s predict as Fa s predict as Re. GR) = 0.012 = 0.206	xe = 103 xe = 494 al = 6					Windows gs to activate Windows	vs.
Ava F TCD 0.0280	•	<b>`</b>								
•							script		Ln 596 C	Col 1

Figure 58: QDA Classification Result of 1000 Fingerprint Images Using 10 IQMs.

lew folder (2) > Abdu code >	Acquistion Control APPS sktop\hp_New folder (2)\Abdu code\ Create_TM_10.m X Class_best_	iology MATLAB Coder (Class_best_10.m 10.m × TM_TEST_10	Application Compiler Dum X unique.m X 1	O     O     Search Documentation     O     Databution     Fiting     +	کر م بر - بر - بر -
System Signal Analysis Memfilication Signal Analysis Levi folder (2) + Abdu code + Signature - C.W.Berk/LENVOY-PC/De bbbjided.mx + hhtm × bbbjided.mx + hhtm × bbbjided.mx + hhtm × bbbjided.mx + hhtm × bbbjided.mx + htm × start = 1000 mm + 10000 mm + 1000 mm + 1000 mm + 1000 mm + 10000  mm + 10000 mm + 10000 mm + 100000 mm + 100000 mm + 100000000 mm + 10000000000	Acquisition instrument SinB Acquisition Control APPS sktoph.hp.hew.folder (2)/Abdu code\ Create_TM_10.m × Class_best_ creance Difference (origing, dis	Ilong XATLAB Coder	Application Classification Compiler Learner	n Distribution Fitting	
Z Editor - C:\User\LENOVO-PC\De bbbjjhdelm × hhhm × 253 254 %TotalCornerDiff 255 - TCD = TotalCorne 256 %disp('15-TotalC	sktop\hp_New folder (2)\Abdu code\ Create_TM_10.m X Class_best_ erence rDifference (origImg, dis-	10.m × TM_TEST_10	0.m × unique.m × ·	+)	• >
Z Editor - C:\User\LENOVO-PC\De bbbjjhdelm × hhhm × 253 254 %TotalCornerDiff 255 - TCD = TotalCorne 256 %disp('15-TotalC	Create_TM_10.m × Class_best_ erence rDifference (origImg, dis	10.m × TM_TEST_10	0.m × unique.m × ·	+ ]	
bbbjjhdel.m × hhh.m × 252 eursp(1rD); 253 254 %TotalCornerDiff 255 - TCD = TotalCorne 256 %disp('15-TotalC	Create_TM_10.m × Class_best_ erence rDifference (origImg, dis	10.m × TM_TEST_10	0.m ≍ unique.m × .	+	_
252 * disp(1rD); 253 254 %TotalCornerDiff 255 - TCD = TotalCorne 256 %disp('15-TotalC	erence rDifference(origImg, dis		0.m × unique.m × ·	+	
253 254 %TotalCornerDiff 255 - TCD = TotalCorne 256 %disp('15-TotalC	rDifference(origImg, dis	t.Tmg):			
260 -         SME = SpectralMa           261 %disp('2-spectra           262 %disp(SME);           263           264 %spectral phase	<pre>gnitudeError(origImg, di: l magnitude error = '); error</pre>	stImg);			
Command Window					Ċ
New to MATLAB? See resources for G	jetting Started.				;
Number of Real images p Number of Real images p 500 Fake fingerprint in Number of Fake images p Number of Fake images p FALSE GENUINE RATE (FFR) = HALF TOTAL ERROR RATE A>>>	predict as Real = 469 predict as Fake = 31 mages result : predict as Fake = 487 predict as Real = 13 ) = 0.026 0.062			Activate Windows	dows
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	155     %spectral magnit       160     SKE     SpectralWatter       161     Wdisp('S-spectral       163     Mspectral phase       164     SEE = SpectralPr       165     SFE = SpectralPr       166     Gammand Window       Number of Real images     Nouber of Real images       100     Pake fingerprint i       100     Number of Fake images       100     Faks       100     Faks       100     Faks       100     Faks	559       supectral magnitude error         601       SHE = SpectralMagnitudeError(origImg, di         611       wdisp('3-epectral magnitude error = ');         612       wdisp(SHE);         613       supectral phase error         614       spectralPhaseError(origImg, disting         615       SFE = SpectralPhaseError(origImg, disting         616       Number of Real images predict as Real = 4669         Number of Real images predict as Pake = 31         600       Fake fingerprint imager serult :         Number of Fake images predict as Pake = 13         FALSE FARE ERTE(FRR) = 0.022         RALF TOTAL ERROR RATE (HTER) = 0.044         % >>	<pre>55 spectral magnitude error 60 SNE = SpectralMagnitudeFror(origImg, disImg); 61 wdisp('2-spectral magnitude error = '); 76 wdisp(SNE); 78 spectralPhase error 78 s = SpectralPhaseFror(origImg, disImg); 78 s = SpectralPhaseFror(origImg,</pre>	<pre>59 supectral magnitude error 61 supectral magnitude error (origing, disting); 61 wdisp('3-espectral magnitude error = '); 63 wdisp(SME); 64 wdisp(CME); 65 supectral phase error 65 supectral phase error 66 supectral phase error 66 supectral phase error 67 supectral phase error 68 supectra</pre>	<pre>59 spectral magnitude error 61 wdisp('3-spectral magnitude error = '); 62 wdisp('3-spectral magnitude error = '); 63 wdisp(SRE); 64 wspectral phase error 65 = SFE = SpectralPhaseError(origImg, distImg); 65 words 66 words 67 words 68 words 60 words 60 words 60 words 60 words 60 words 61 words 62 words 63 words 64 words 64 words 64 words 64 words 65 words 66 words 66 words 66 words 66 words 66 words 66 words 67 words 66 words 67 words 67 words 67 words 68 words 60 words 6</pre>

Figure 59: Linear SVM classification result of 1000 fingerprint images using 10 IQMs.

## 4. Quadratic SVM

HOME PLOTS APPS EDITOR	PUBLISH VIEW	🦻 🖻 🖻 🕐 Search Documentation 🛛 🔎
Image: Second secon	BREAKPOINTS RUN	•
Current Folder	Zeditor - C:\Users\LENOVO-PC\Desktop\hp New folder (2)\Abdu code\Class_best_10.m	•
Name A	bbbjijhdel.m X hhh.m X Create TM 10.m X Class best 10.m X TM TEST 10.m X unique.m X	
addu finger database Infiret JANDAm AdduLm	<pre>66 %disp(MSE); 67 68 %spectral magnitude error 69 ~ SME = SpectralMagnitudeError(origImg, distImg); 70 %disp('2-spectral magnitude error = '); 71 %disp(SE); 72 73 %spectral phase error 74 ~ SFE = SpectralPhaseError(origImg, distImg); 75 %disp('3-spectral phase error = '); 76 %disp(SEE); 77 %gradient magnitude error 78 %gradient magnitude error 79 %gradient magnitude error</pre>	
letails ^	Command Window	
Vorkspace 🐨	New to MATLAB? See resources for Getting Started.	
Jame A         Value           uns         1           Arug F, GPE         0.4287           Arug F, FMO         0.2221           Arug F, FNAE         0.2270           Arug F, FAME         0.2221           Arug F, FAME         0.2221           Arug F, FAME         0.2020           Arug F, FAME         0.0020           Arug F, FSMR         2.0302           Arug F, SFNR         2.03120           Arug F, SFNR         2.13201	S00 real fingerprint images result : Number of Real images predict as Real = 478 Number of Real images predict as Real = 22 S00 Fake fingerprint images result : Number of Fake images predict as Real = 481 Number of Fake images predict as Real = 19 FALSE GENUME RATE(FRG) = 0.038 FALSE FARE RATE(FFR) = 0.044 M HalF TOTAL ERGOR RATE (HTER) = 0.041	Activate Windows Go to Settings to activate Windows

Figure 60: Quadratic SVM classification result of 1000 fingerprint images using 10 IQMs.

HOME     PLOTS     APPS     EDITOR     PUBLICH     VEW     Image: The state of the state o	tion 🔎
Out North Package       Curve Filing       Optimization       PD Tuner       System       System       System       System       System       System       System       Control       Control       Compare       Learner       Disk-dooin       Disk-dooin <thdisk-dooin< th="">       Disk-dooin       <thdisk-dooin< th="">       Disk-dooin</thdisk-dooin<></thdisk-dooin<>	
<pre></pre>	
Current Folder	
Name *     *     bbbjindedm ×     Nhhm ×     Create_TM4_10m ×     Th4_TEST_10m ×     Unique.m ×     +       in hog     finge database     01     % disp (GME);     03     *     *       in hog     Abd.     02     % gradient phase error     03     % gradient phase error     *       in bdu/m     05     % disp (CFE);     03     % disp (CFE);     *     *       in bdu/m     05     % disp (CFE);     05     % disp (CFE);     *       in bdu/m     05     % disp (CFE);     *     *       in bdu/m <td>τ.</td>	τ.
<pre>abdu</pre>	0
0         figge database         02           0         hiftest         03           0         hiftest         03           0         bdum         04         GEF = GradientphaseError           0         bdum         05         bdigradiate           0         afderdencem         05         bsignal to noise ratio           0         bbightedm         50         SIR = SignalcoloseRatio(origImg, disTmg);	
a) Mifed     33     %gradient phase error       3ANDAm     84     GFE     GradientphaseError(origImg, distImg);       3abdum     85     %disp(fSE);       3abdum     86     %disp(GFE);       3abdum     86     %disp(GFE);       3abdum     88       3atudercom     89       4bdus/ferecem     89       5     %sign(1=0)(origImg, distImg);       5     %sign(1=0)(origImg, distImg);	^
AbdUm         84 -         GPE - GradientphaseError(origImg, distImg);           abdum         85         bdimp('S-gradientphaseError(origImg, distImg);           abdufam         86         bdimp('S-gradientphaseError = ');           abdum         86         bdimp('S-gradientphaseError = ');           abdum         87         bdimp('S-gradientphaseError = ');           afstudfencem         87         bdimp('S-gradientphaseError = ');           bdum/defencem         87         bdimp('S-gradientphaseError = ');           bdum/defencem         85         bsignal to noise ratio           bbbbjeddm         50 -         SNR = SignalcohoiseKatio(origImg, distImg);	
Adulam     65     012     012     012     012     012       Adulam     65     %lisp('5-gradient phase error = ');       AdulawedByTabrizm     66     %lisp('6FE);       AdulawedByTabrizm     87       atterdefercem     68       AtterageDfferenem     59 -       SNR = Signalto(origImg, distImg);	
BobJuAn         Bf         %disp(GFE);           AbduSvet@Jlabizm         67           BobJuAn         88           #sterdeforcam         89           %signal to noise ratio         SNR = SignalcoNoiseRatio(origImg, distImg);           bbbjhdelm         50	
AbdushwedByTabrizm     67       abduum     68       Arterdefencem     68       AvagaDfferencem     69       bbbjmdelm     90 -       SNR = SignalcoNoiseRatio(origImg, distImg);	
Bobuum         68           #ferdefencem         89           Assagbiffencem         90 -           SNR = Signalto(origing, disting);	
AverageDifference.m     89     % signal to noise ratio       AverageDifference.m     90 -     SNR = SignaltoRoiseRatio(origImg, distImg);       bbbjmdeLm     90 -     SNR = SignaltoRoiseRatio(origImg, distImg);	
MexageOffrence.m         90 -         SNR = SignaltoNoiseRatio(origImg, distImg);           bbbjmdelm         90 -         SNR = SignaltoNoiseRatio(origImg, distImg);	
Dbbjjndel.m	
Desty_classincation.m     92 %disp(SNR);	
A bartfourte management 93	
	````
Details Command Window Command Window	- (
Workspace 🐨	
Name  Value New to MATLAB? See resources for <u>Getting Started</u> .	
ans 1 500 real fingerprint images result :	
Number of Real images predict as Real = 483	
Avg FMD 0.221 Number of Real images predict as Fake = 17	
Avg F NAE 0.0270 500 Fake fingerprint images result :	
Ang.F.PANE 70201 Number of Fake images predict as Fake = 460 Ng.F.PANE 0.0001 Number of Fake images predict as Fake = 40	
AVg_r_NAMD 0.2002	
AVg_F_SWE 0.0049	
ACLIVATE TATAL EDDOD DATE (UTED) = 0.057	
Avg FCO 0.0280 V C nDr TOTAL ERKOR RATE (n1EK) = 0.057 Go to Settings to activate W	/indows.

Figure 61: Fine KNN classification result of 1000 fingerprint images using 10 IQMs.

номе	PLOTS APPS	EDITOR	PUBLISH	VEW						4 4 5	€ 5 3	Search Documentat	tion 🔎
et More Install Pa	ckage Curve Fitting	Optimization PID Tu		Signal Analysis	Image Acquisition	Instrument Control	SimBiology	MATLAB Coder	Application Compiler	Classification Learner	Distribution Fitting	·	
FILE					,	VPPS							
🔶 🖬 🔁 📒	► C: ► Users ► LENOVO	D-PC 🕨 Desktop 🕨 h	_New folder (2) 🕨	Abdu code 🕨									
urrent Folder		۲	📝 Editor - C:\U	ers\LENOVO-PC\	Desktop\hp_Nev	v folder (2)\Abdu	code\Class_l	best_10.m					0
Name +			bbbiihdel.n	×   hhh₊m ⇒	Create TM	10.m × Clas	best 10.m	X TM TEST 10	).m ≍Ìuni	aue.m ×]+			
abdu finger dataas hiftiest abdu.m babdu.m babdu.m babdu.m babdu.m babdu.m babdu.m bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific bests.classific	Tabriz.m n ence.m		255         TC           256         %d           257         %d           258         %s           259         %s           260         SM           261         %d           262         %d           263         %d           264         %s           265         SP	<pre>btalCornerDi D = TotalCorn isp('15-Tota isp(TCD); bectral magn Z = Spectral isp('2-spect isp(SME); bectral phass Z = Spectral isp('3-spect</pre>	nerDifferen lCornerDiff itude error MagnitudeEr ral magnitu e error PhaseError (	erence = ') ror(origImg de error = forigImg, di	; , distImç ');						
orkspace		(7)	Command Wind	ow									
ame ▲ ans Avg_F_GPE Avg_F_MD Avg_F_NAE Avg_F_PSNR Avg_F_SNR Avg_F_SNE Avg_F_SNR Avg_F_SPE Avg_F_SPE Avg_F_CD	Value 1 0.4287 0.2221 0.0270 79.2081 0.2002 0.0049 28.1320 148.9251 0.0280	· · · · · · · · · · · · · · · · · · ·	500 real Number o: S00 Fake Number o: Number o: FALSE GEI FALSE FAI	87 See resources for fingerprint real images fingerprint rake images rake images UINE RATE (FCR) L ERROR RATE	images res s predict a images res s predict a s predict a s predict a GR) = 0.052 = 0.088	ult : s Real = s Fake = ult : s Fake = s Real =	456 44 474 26					Windows Igs to activate V	Vindows.

Figure 62: Logistic Regression classification result of 1000 fingerprint images using 10 IQMs.

# 5. Table and Screenshot of Training Results For 100 Fingerprint Images Using 5 IQMs

100 fingerprint images with 5 IQMs result using 6 classifiers:

## 1. LDA

HOME PLOTS	APPS	EDITO	DR	PUBLISH	VIEW						1 H D	🖻 🔁 🕄 Sea	irch Documenta	tion	P
t More Install Package Curve		Optimization	MuPAD Notebook	PID Tuner	System Identification	Signal Analysis	Image Acquisition	Instrument Control	SimBiology	MATLAB Coder	Application Compiler	Classification Learner	•		
FILE						A	PPS								
🔶 🛅 💭 📙 🕨 C: 🕨 Users	<ul> <li>LENOVC</li> </ul>	D-PC 🕨 Deskte													
rrent Folder						p\hp_New folder (		TMTEST_5.m							C
Name 🔶			TM_5	i_Create.m →	TM_Best_5.n	n × TMTEST_5	.m × +								
abdu		^	557												
finger database			558			MSE-Avg_F_MS			R, Avg_R_P	SNR-Avg_F_PS	NR, Avg_R_1	K-Avg_F_NK,	Avg_R_AD-Av	'g_F_AD, A	٤vg
results 3AND4.m			559 -			SNR.';'RAMD.									
abdu.m			560 -	best5=[di %S	f_btwn_SNE	_FSNR;dif_b1	wn_PSNR_F	SNR;dif_bt	wn_RAMD_FI	RAMD; dif_btw	n_SME_FSMI	;dif_btwn_G	PE_FGPE];		
abdu4.m			561 562 -		ble(title,										
AbduSavedBvTabriz.m			562 -			.Dest5)	TOM DANK								
abduu.m			564 -			LA, ('best5'),			10de1)						
afterdefence.m			565	CDID 🚆 SC	TCTOM8(CD1	LA, ( Dests ),	( · descend								
AverageDifference.m			566 -	disp(rr);											
bbbjjhdel.m			567 -	disp(II);											
best5_classification.m			568 -	disp(rrl											
best5wrk.asv best5wrk.m			569 -	disp(rfl											
best10wrk.m			570 -	disp(FGR	);										
		~	571 -	disp (FFR	);										
ils		^	1 K												>
kspace		۲													-
ne 🔺 Value			Comman	d Window											
ns 1		^													
vg_F_GPE 0.4468		^				dict as Real									
vg F PSNR 79.5718						dict as Fake									
vg_F_RAMD 0.1901						dict as Fake									
vg_F_SME 142.4203				er of Fake GENUINE RA		dict as Real	= 18								
vg_F_SNR 28.5125				FAKE RATE (											
wg_R_GPE 0.4857				TOTAL ERRO											
Vg_R_PSNR 78.0187			fx >> =	IVIAL ERRO	N NAIL (DI	ER/ - 0.25									
Avg_R_RAMD 0.2226		~	<												
19911208											script			592 Col	
											scripe		Ln	352 COI	2 D

Figure 63: LDA training result of 100 fingerprint images using 5 IQMs.

MATLAB R2016a	PLOTS APPS	EDITOR	PUBLISH	VIEW					8 6 .	4 <b>6 6</b> 5	. € 6 0	 Search Documentatio	× 10 م م
Get More Install Pack Apps App Ap	age Curve Fitting O	ptimization PID T	-		Image Acquisition	Instrument Control	SimBiology	MATLAB Coder	Application Compiler	Classification Learner	Distribution Fitting	•	
FILE					A	PPS							
	C: • Users • LENOVO-												-
Current Folder			1.00	C:\Users\LENOVO-PC\[									$\odot$
Asme A     Asme A	ice.m		<ul> <li>262</li> <li>263</li> <li>264</li> <li>265</li> <li>266</li> <li>267</li> <li>268</li> <li>269</li> <li>270</li> <li>271</li> <li>272</li> <li>273</li> <li>274 -</li> </ul>	Create.m × TMTES *spectral phase *SPE = 0; *disp('3-spectr) *disp('SPE); *gradient magn: *GME = 0; *disp(GME); *gradient phase GFE - Gradientp	e error itude error ent magnitu e error phaseError(	rror = '); de error = origImg, di		ification.m 🔀	best5wrk.m	≍ Best_10.r	m ≍ (bbbjjho	felm ≍ [ + [	
bbjjhdel.m (Script)		^											,
Vorkspace		0	)	Window ATLAB? See resources fo									
Name A Avg.F.GPE Avg.F.RAMD Avg.F.SNR Avg.F.SNR Avg.F.SNR Avg.R.GPE Avg.R.PSNR Avg.R.PSNR Avg.R.PSNR	Value 1 0.4468 79.5718 0.1952 142.4203 28.5125 0.4937 77.3407 0.2297		Numbe Numbe Numbe FALS FALS	r of Real images r of Real images r of Fake images genuine RATE (FGF FAKE RATE (FFR) = TOTAL ERROR RATE	<pre>predict a: predict a: predict a: predict a: predict a: 0 = 0.36 0.12</pre>	s Real = s Fake = s Fake = s Real =	44 6 32 18					Windows	ndour
Ava R SME	224.5596		< <										>
										script		Ln	539 Col 3

Figure 64: QDA training result of 100 fingerprint images using 5 IQMs.

Oct More hash Pacage       Ource FEng Optimization       PD Tuner       System Spraid Analysis       Tage Analysis	StriBiology MATLAB Coder Application Classification Distribution
Image: Solution of the second seco	Compiler Learner Fitting
ument Folder       Image: Sector	
Name *       abdu       TMS_Counter X       TMEST_5m × TM_Backs         abdu       55       Wean Square Error       TMST_5m × TM_Backs         JabduAm       55       Step Step Step Step Step Step Step Step	-
bidu     55     Whean Square Error       ingr disbase     55     Whean Square Error       ingr disbase     55     Whean Square Error       indudum     55     Wiles - 0;       inductor     56     Wiles - 0;       inductor     57     Wile - 0;       inductor     56     Wiles - 0;       inductor     56     Wiles - 0;       inductor     56     Station, Norther of the Norther of	
56     Winean Square Error       MADUAm     S6       Matagodificencem     S6       Matagodificencem     S6       Matagodification.m     S6       Matagodification.m     S6       Motspace     S6       Norkspace     S6       Norkspace     S6       Name ^     Value       Name ^     Value       Name ^     Value	× best5_classification.m × best5wrk.m × Best_10.m × bbbjjhdel.m × +
Workspace         Command Window           Name A         Value         New to MATLAB? See resources for <u>Getting Started.</u> Ins 1         Number of Real images predict as Real           Workspace         Audes	img, distImg);
Vorkspace v v Name A Value New to MATLAB? See resources for <u>Getting Started</u> . Ans 1 Aygr_GPE 0.4468 Number of Real images predicts a Real	
Avg_F_GPE 0.4468 Number of Real images predict as Real	
Ward F, SAMO         (5.) 10           Ward F, SAMO         0.1552           Ward F, SAMO         0.1552           Ward F, SAMO         142.4003           Ward F, SAMO         285.155           Ward F, SAMO         285.155           Ward F, SAMO         0.397           Ward F, SAMO         0.397           Ward F, SAMO         0.2277           Ward R, SAMO         0.2277           Xard R, SAMO         0.22576	35

Figure 65: Linear SVM training result of 100 fingerprint images using 5 IQMs.

# 4. Quadratic SVM

📣 MATLAB R2016a										_	ø ×
HOME PL	LOTS APPS	EDITOR	PUB	LISH VIEW					69660	Search Documentation	ء م
Get More App	ge Curve Fitting	Optimization F		System Signal Analysis	Image Inst Acquisition C	rument SimBiology ontrol	MATLAB Coder	Application Class	ification Distribution arner Fitting	•	
FILE 🗲 🔶 🔁 🔽 🚺 💽				ler (2) 🕨 Abdu code 🕨	APPS						<del>-</del> ,
· · · — •	C: • Users • LENOV	/O-PC • Desktop				(0)					12
Current Folder				or - C:\Users\LENOVO-PC\[							; ⊙
Name A add Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu Addu	e.m n.m		<ul> <li>70</li> <li>71</li> <li>72</li> <li>73</li> <li>74</li> <li>75</li> <li>76</li> <li>77 -</li> <li>78</li> <li>79</li> <li>80</li> <li>81</li> <li>82</li> <li></li> </ul>	<pre>ts_createm x/ MTES terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport, terrsport</pre>	itude error ent magnitude e e error phaseError(orig ent phase error	<pre>rror = '); (Img, distImg);</pre>	sification.m X	bestŠwrk.m 🗶 🛛 Br	st_10.m × bbbjjhd	elm × +	~
			C	and Window							(
Workspace			New to	MATLAB? See resources fo	r Getting Started						
Ans     Avg_F_GPE     Avg_F_SNR     Avg_F_RAMD     Avg_F_SME     Avg_F_SNR     Avg_F_SNR     Avg_R_GPE     Avg_R_PSNR     Avg_R_RAMD	Value 1 0.4468 79.5718 0.1952 142.4203 28.5125 0.4937 77.3407 0.2297		A Nur Nur Nur FAI FAI FAI FAI	aber of Real images aber of Real images aber of Fake images aber of Fake images S GENUINE RATE(FGF S FAKE RATE(FFR) = .F TOTAL ERROR RATE	<pre>predict as Re predict as Fa predict as Fa predict as Re predict as Re 0 = 0.16 0.12</pre>	ke = 6 ke = 42					,
	224.5596		× <								>
								script		Ln 588	Col 57

Figure 66: Quadratic SVM training result of 100 fingerprint images using 5 IQMs.

HOME PLOTS APPS EDI	TOR	PUBLISH	VEW					<b>B H</b> .	4 <b>L</b> 3	∉ ⊡ ?	Search Documentation	Q
et More Instal Package App App Curve Fitting Optimization	PID Tur	er System Identification	Signal Analysis	Image Acquisition	Instrument Control	SimBiology	MATLAB Coder	Application Compiler	Classification Learner	Distribution Fitting	•	
FILE					APPS							
🕩 🖬 🛜 📙 🕨 C: 🕨 Users 🕨 LENOVO-PC 🕨 Desi												-
urrent Folder	۲	Z Editor - C:\Use				3						$\odot$
Name 4 abdu finger database bilitest 3 AND4 m 2 abduum 2 abd		251         *011           252         %di           253         254           255         SME           256         %di           257         %di           259         %sp           259         %sp           261         %di           262         %di           263         %di	<pre>sp('IS-IOLA ap(TCD); ectral magn = Spectral ap('2-spect ap(SME); ectral phas 5 = 0;</pre>	itude erro MagnitudeE ral magnit e error ral phase	error = ');	g, distImç ');		best5wrk.m	× Best_10.m	≍ (bbbjjhd	feLm × +	
tails	~	<										>
orkspace	()	Command Windo	w									
		New to MATLAB	See resources fo	or Getting Start	ed.							
wine         Value           avg.f.gpE         0.448           Avg.f.pSNR         79.5718           Avg.f.PSNR         0.152           Avg.f.PSNE         142.403           Avg.f.SNE         28.5125           Avg.f.SNE         0.4837           Avg.R.PSNR         77.3407           Avg.R.PSNR         0.2237           Avg.R. SNE         2245596	^	Number of Number of FALS GENU FALS FAKE	Real image Real image Fake image Fake image NE RATE (FG RATE (FFR) . ERROR RAT	s predict s predict s predict R) = 0 = 0	as Fake = as Fake = as Real =	50 0 50 0						
AVG K SME 224.3390											1 501	
									script			1 Col 1 7 PM

Figure 67: Fine KNN training result of 100 fingerprint images using 5 IQMs.

📣 MATLAB R2016a													-	ø ×
HOME P	LOTS APPS	EDIT	DR	PUBLISH	VEW					26.	4 6 6 9	🖻 🔁 🕐 Se	arch Documentatio	n 🔎
Get More Install Packa Apps App App	ge Curve Fitting	Optimization	PID Tuner	System Identification	Signal Analysis	Image Acquisition	Instrument Control	SimBiology	MATLAB Coder	Application Compiler	Classification Learner	Distribution Fitting	•	
FILE							APPS							
🗢 🔶 💽 🎘 📙 🕨	C: • Users • LENO	VO-PC   Deskt	op ► hp_t	New folder (2) 🕨	Abdu code 🕨									· · .
Current Folder			۲	📝 Editor - C:\Us	ers\LENOVO-PC\	Desktop\hp_Ne	w folder (2)\Abo	du code\TMTE	ST_5.m					$\odot$
🗋 Name 🔺				TM_5_Creat	e.m × TMTE	ST_5.m ≍ T	M_Best_5.m 💚	best5_class	$;$ ification.m $\times$	best5wrk.m	× Best_10.m	🛛 🛛 🕹 bbbjjhdel.	m ≍[+]	
abdu     bdu     finger database     hiftest     AbduAm     abdu4m     a	e.m		×	77 - GPE 78 %di 79 %di 80 81 82 %si 83 - SNR 84 %di 85 %di 86 87 %Pe	adient phas = Gradient sp('5-gradi sp(GPE); gnal to noi := Signalto sp('6-signa sp(SNR); ak Signal t R = PeakSig	phaseError ent phase ( se ratio NoiseRatio to to noise o Noise Rat	(origImg, c e ratio = '	listImg); );	mg);					~
Details			^	<										>
Workspace			۲	Command Winde										
Name  ans avg_F_GPE Avg_F_SNR Avg_F_SME Avg_F_SNR Avg_F_SNR Avg_R_GPE Avg_R_PSNR Avg_R_AMD	Value 1 0.4468 79.5718 0.1952 142.4203 28.5125 0.4937 77.3407 0.2297			Number of Number of FALS GENU FALS FAKE HALF TOTA fx >> 6	Real image: Real image: Fake image: Fake image: INE RATE (FG RATE (FFR)	s predict a s predict a s predict a s predict a R) = 0.2 = 0.12	s Real = s Fake = s Fake = s Real =	44 6 40 10						,
Ava R SME	224.5596		×	<										
											script		Ln 5	94 Col 20

Figure 68: Logistic Regression training result displayed for 100 fingerprint images using 5 IQMs.

# 6. Screenshot of Classification Results For 1000 Fingerprint Images

# Using 5 IQMs:

1000 fingerprint images with 5 IQMs result using 6 classifiers:

## 1. LDA

HOME PLOTS APPS I	DITOR PUBLISH VIEW 🚺 🖥 🖉 🐨 🔁 🕐 Search Documentation	P
New New Open Compare Import Save Data Workspace	New Variable     Analyze Code     Image: Constraint of the second	
	IABLE CODE SIMULINK ENVIRONMENT RESOURCES	-
urrent Folder	Compose of the second sec	
	Control - Croster LENOVO-P Close Statistical provided Concerned Lassing and the statistical provided Concerned Lassing	٢
finguré database     finguré database     finguré database     finguré     ANUA m     ANUA m     douburm     douburm     douburm     douburm     finguré database     finguré des database     finguré des database     finguré des database     betrofference m     betrofference m	<pre>542 - disp(tructownir, 543 - disp(tructownir, 544 - disp(tructowni); 544 - disp(tructowni); 545 - grammatrix disp(f30) real fingerprint images result :'); 546 - grammatrix disp(f3) real fingerprint images predict as Fake = ',num2str(fakecount)]; 547 - disp(f7); 548 - disp(f7); 559 - rif= ('Number of Fake images predict as Fake = ',num2str(fakecount)]; 551 - rif= ('Number of Fake images predict as Rake = ',num2str(fakecount)]; 552 - disp(f7); 553 - disp(f7); 553 - disp(f7); 554 - disp(f7); 555 - FFRe('FALSE CHOUNE FART(fFR) = ',num2str(fruecount/f300)]; 555 - FFRe('FALSE CHOUNE FART(fFR) = ',num2str(fakecount/f300)]; 556 - FFRe('FALSE CHOUNE FART (FFR) = ',num2str(fakecount/f300)]; 557 - FFRe('FALSE CHOUNE FART (FFR) = ',num2str(fakecount/f300)]; 558 - FFRe('FALSE CHOUNE FART (FFR) = ',num2str(fakecount/f300)]; 559 - FFRe('FALSE CHOUNE FART (FFR) = ',num2str(fakecount/f300)]; 550 - FFRe('FALSE CHOUNE FART (FFR) = ',num2str(fakecount/f300)]; 551 - FFRe('FALSE CHOUNE FART (FFR) = ',num2str(fakecount/f300)]; 552 - FFRe('FALSE CHOUNE FART (FFR) = ',num2str(fakecount/f300)]; 553 - FFRe('FALSE CHOUNE FART (FFR) = ',num2str(fakecount/f300)]; 554 - FFRe('FALSE FART (FFR) = ',num2str(fakecount/f300)]; 555 /pre>	, ,
Iteme ▲         Value           Gaustif         480x400 double           GPE         1000h1 double           H         1001h1 double           H         500 for the	CommandWindow Number of Real images predict as Real = 417 Number of Real images predict as Fake = 03 Number of Fake images predict as Fake = 336 Number of Fake images predict as Fake = 3164 FALSE GRUVINE RATE (FICF) = 0.328 FALSE TARE FATE(FICF) = 0.328 FALSE TARE FATE(FICF) = 0.326 FALSE TOTAL ERROR RATE (HTER) = 0.247 A >>	

Figure 69: LDA classification result of 1000 fingerprint images using 5 IQMs.

HOME PLOTS APPS EDITOR	PUBLISH VIEW	🔓 🖻 🗇 🗭 🔁 🕐 Search Documentation 🛛 🔎
	) 🐼 🐠 🚺 🔛 🖉 🚟	🚮 🔼 🛓
		Classification Distribution
Apps App App		Learner Fitting
FILE	APPS	
= 🔶 🛅 💭 📙 🕨 C: 🕨 Users 🕨 LENOVO-PC 🕨 Desktop		-
urrent Folder	📝 Editor - C:\Users\LENOVO-PC\Desktop\hp_New folder (2)\Abdu code\best5wrk.m	۲
Name 🔺	TM_5_Create.m X TMTEST_5.m X TM_Best_5.m X best5_classification.m X best5wrk.m X	Best_10.m × bbbjjhdel.m × +
abdu	190	
finger database	199	
hlfitest	200 %Mean Square Error	
1 3AND4.m	201 %MSE = 0;	
🚹 abdu.m	<pre>202 %disp('1-Mean Square Error = ');</pre>	
🛃 abdu4.m	203 %disp(MSE);	
魡 AbduSavedByTabriz.m	204	
🔮 abduu.m	205 %signal to noise ratio	
魡 afterdefence.m	206 - SNR = SignaltoNoiseRatio(origImg, distImg);	
AverageDifference.m	<pre>207 %disp('6-signato to noise ratio = ');</pre>	
魡 bbbjjhdel.m	208 %disp(SNR);	
best5_classification.m	209	
e best5wrk.asv	210 %Peak Signal to Noise Ratio	
best5wrk.m	211 - PSNR = PeakSignaltoNoiseRatio(origImg, distImg);	
bbjjhdel.m (Script)	<	>
Vorkspace	Command Window	
Value Value	New to MATLAB? See resources for Getting Started.	
ans 1	500 real fingerprint images result :	
Avg_F_GPE 0.4287	Number of Real images predict as Real = 364	
Avg_F_SNR 79.2081	Number of Real images predict as Fake = 136	
Avg F RAMD 0.2002	500 Fake fingerprint images result :	
Avg_F_NAMD 0.2002 Avg_F_SME 148.9251	Number of Fake images predict as Fake = 339	
Avg_F_SNR 28.1320	Number of Fake images predict as Real = 161	
Avg R GPE 0.5488	FALSE GENUINE RATE(FGR) = 0.322	
Avg R PSNR 77.9648	FALSE FAKE RATE(FFR) = 0.272	Activate Windows
Avg_R_RAMD 0.2295	HALF TOTAL ERROR RATE (HTER) = 0.297	Go to Settings to activate Windows.
	fx >>	

Figure 70: QDA Classification of 1000 Fingerprint Images Using 5 IQMs.

📣 MATLAB R2016a										- 0	×
номе	PLOTS APPS	EDITO	R PU	IBLISH VIEW					🖢 🚖 🖨 🕐 Searc	h Documentation	<mark>م</mark> 2
Get More Install Pa Apps App	ckage App	Optimization	PID Tuner	System Signal Analysis Identification	Image Instrume Acquisition Control		MATLAB Coder	Application Classifi Compiler Lear	cation Distribution	·	
FILE	A C A User A USE		a b ha blaufa	older (2) 🕨 Abdu code 🕨	APPS						- ,
Current Folder	C: V USERS V LEINU	VO-PC / Deskto		litor - C:\Users\LENOVO-PC\I	Dealers and the Marco Califord (2)	Aboli and Abore					Ô
Name A					ST 5.m × TM Best 5.n		_	best5wrk.m × Best	t 10.m × bbbijhdel.m		U U
abdu finger databas hfirest 3ANDAm abdum AbdusavedBy abduum AverageDiffer bbijhdelm best5.classific best5wrkm best5wrkm	Tabriz.m n ence.m		<ul> <li>190</li> <li>199</li> <li>201</li> <li>202</li> <li>203</li> <li>204</li> <li>205</li> <li>206</li> <li>207</li> <li>208</li> <li>209</li> <li>210</li> <li>211</li> </ul>	<pre>%Mean Square E: %MSE = 0; %disp('1-Mean : %disp(MSE); %signal to noi; SNR = Signaltol</pre>	rror Square Error = '); se ratio NoiseRatio(origImg to to noise ratio	, distImg);					~
Details				mand Window							
Norkspace				to MATLAB? See resources fo	r Gatting Started						
Name A Ans Avg_F_GPE Avg_F_PSNR Avg_F_RAMD Avg_F_SME Avg_F_SNR Avg_R_GPE Avg_R_OPE Avg_R_AND	Value 1 NaN NaN NaN NaN NaN 0.5488 77.9648 0.2295		A No	umber of Real images umber of Real images umber of Fake images umber of Fake images ALSE GENUINE RATE(FC ALSE FAKE RATE(FFR) ALF TOTAL ERROR RATE >	s predict as Real s predict as Fake s predict as Fake s predict as Real SR) = 0.334 = 0.104	= 52 = 333					
Ava R SME	199.7606		× <								>
								script		Ln 579 C	ol 1

Figure 71: Linear SVM LDA classification result of 1000 fingerprint images using 5 IQMs.

# 4. Quadaratic SVM

Image: Package       Curve Fiting       Image: Package       Curve Fiting       Image: Package       Image: Pa	MATLAB R2016a	PJELSH VEW	o x
<pre></pre>	et More Instal Package Curve Fitting Optimization PD	uner System Signal-Ansiysis Image Instrument Smölbogy MATLAB Coder Applaation Distribution	
Current Folder Current Current Folder Current Current Folder Current Current Folder Current Current Current Cur			
Nmme ^       Value         MADUM       MAS_Creater X MMESIAm X MMESIAm X betjakasficationm X betjakasfication X betjakasfi X bispiritin X betjakasf	= 🕪 💽 🔀 📙 🕨 C: 🕨 Users 🕨 LENOVO-PC 🕨 Desktop 🕨		•
a       Adu       7       Yeradient phase error         b       figwrddiaws       91       Yeradient phase error       91         b       Adum       91       GPE = GradientphaseError(origImg, distImg);       91         b       Adum       91       GPE = GradientphaseError(origImg, distImg);       91         b       Adum       91       GPE = GradientphaseError(origImg, distImg);       91         b       Adum       91       Sigmat Conject and Sigmat C	urrent Folder	Z Editor - C:\Users\LENOVO-PC\Desktop\hp_New folder (2)\Abdu code\best5wrk.m	$\odot$
ange database     00     synchicit phase error       billion     00     synchicit phase error       bibly     00     00       b	Name 🔺	TM_5_Create.m X TMTEST_5.m X TM_Best_5.m X best5_classification.m X best5wrk.m X Best_10.m X bbbjjhdel.m X +	
Norkspace         Command Window           Name A         Value         New to MATLABS See resources for <u>Gatting Stanted</u> .           ns         1         Mag. F.GPE         0.4287           way, F.PSR         0.4287         Number of Real images predict as Real = 477           way, F.PSR         0.4287         Number of Real images predict as Real = 477           way, F.SAND         0.2002         Number of Real images predict as Real = 477           way, F.SANR         28.120         Number of Fake images predict as Real = 360           way, F.SANR         28.120         Number of Fake images predict as Real = 140           way, F.SANR         7.5468         FALSE FARE RATE (FRFR = 0.048           way, R.PSANR         7.2054         FALSE FARE RATE (FRFR) = 0.163	frigger database hiffrest abdum abdum abduk-m AbduS-wetByTabrizm abduum abduum abduum abduum abduum abduum abduse betBytBum betBytBum betBytBum betBytBum betBytBum betBytBum betBytBum betBytBum betBytBum betBytBum betBytBum betBytBum betBytBum betBytBum betBytBum betBytBum betBytBum betBytBum betBytBum betBytBum betBytBytBum betBytBytBytBytBytBytBytBytBytBytBytBytByt	0     %gradient phase error       11 -     GFE = GradientphaseTror(origImg, distImg);       22     %disp('5-gradient phase error = ');       83     %disp('5-gradient phase error = ');       84     %disp('5-gradient phase error = ');       85     %disp('s-gradient phase error = ');       86     %eignal to noise ratio       87     SNR = SignaltoNoiseRatio(origImg, distImg);       88     %disp('6-signalto to noise ratio = ');       89     %disp(SNR);       90     %disp(SNR);       91     %Peek Signal to Noise Ratio       92     Puter production encodering distImplication d	~
Workspace         O           Name A         Value           Anne A         Value           Name A         Value           Ang F, GSFE         0.4287           May F, FSAR         73.0001           May F, FSAR         23.0002           May F, FSAR         28.1300           May F, FSAR         28.1300           May R, FSAR         77.5648           May R, FSAR         70.2681           May R, FSAR         70.4681           May R, FSAR         70.4681           May R, FSAR         70.4681           May R, FSAR         70.4681           May R, FSAR         70.412 FRAR DATE (FTRF) = 0.163		Common Differ from	
Name     Value     500 real fingerprint images result:       ans     1       Ang.FGFR     0.427       Mays.FSR     0.200       Mays.FSR     28.120       Mays.FSR     28.120       Mays.FSR     0.5488       Mays.FSR     77.5648       Mays.FSR     77.5648       Mays.FSR     77.255       Mays.FSR     77.5648	/orkspace	)	
H Ava R SME 199.7606	ans 1 Arg_F_GPE 0.4287 Arg_F_PSNR 73,2081 Arg_F_RAMD 0.2002 Arg_F_SNR 1458,251 Arg_F_SNR 28,1320 Arg_F_GPSNR 0.5488 Arg_R_PSNR 77,5648 Arg_R_RSNR 0.2255	500 real fingerprint images result :         Number of Real images predict as Real = 477         Number of Real images predict as Real = 23         500 Fake fingerprint images result :         Number of Fake images predict as Fake = 360         Number of Fake images predict as Real = 140         FALSE CRUNTE RATE(FR) = 0.28         FALSE FAKE RATE(FFR) = 0.046         HALF TOTAL ERROR RATE(FTR) = 0.163	
script Ln 591 Col 5			

Figure 72: Quadaratic SVM classification result of 1000 fingerprint images using 5 IQMs.

MATLAB R2016a		- ø ×
HOME PLOTS APPS EDITOR	PUBLISH VEW	ntation 🔎 💈
Apps App App	iner System SgnalAnalysis mage Instrument SmBiology MATLAS Coder Appleaton Classification Distribution Complex Fitting	
FILE	APPS	<del>,</del> -
Current Folder	np_vew folder (2) + Abdu Code +	• ×
		. ⊜×
🗋 Name 🔺	TM_5_Create.m X TMTEST_5.m X TM_Best_5.m X best5_classification.m X best5wrk.m X Best_10.m X bbbijhdel.m X +	
abdu	<pre>234 stup = 0; 234 stupp('ll-Maximum Difference = ');</pre>	^
■ finger database	235 Voltey (II) :	
hlfitest     Alphan     Alphan	236	
abdum	237	
abdu.m	238 %RaverageMD	
AbduSavedBvTabriz.m	239 - RAMD = RarverageMD(origImg, distImg);	
abduu.m	240 %disp('12-RaverageMD = ');	
afterdefence.m	241 %disp(RAMD);	
AverageDifference.m	242	
🕙 bbbjjhdel.m	243 Normalized Absolute Error	
魡 best5_classification.m	244 %NAE =0;	
魡 best5wrk.m	<pre>245 %disp('13-Normalized Absolute Error = ');</pre>	
best10wrk.m	246 %disp(NAE);	~
Details	<	>
Workspace	Command Window	(
Name 🗠 Value	New to MATLAB? See resources for Getting Started.	
ans 1		
Avg_F_GPE NaN	Number of Real images predict as Real = 443	
Avg_F_PSNR NaN	Number of Real images predict as Fake = 57	
Avg_F_RAMD NaN	Number of Fake images predict as Fake = 343	
Avg_F_SME NaN	Number of Fake images predict as Real = 157 FALSE GENUINE RATE(FGR) = 0.314	
Avg_F_SNR NaN	FALSE GENUINE RATE (FGR) = 0.314 FALSE FAKE RATE (FFR) = 0.114	
Avg_R_GPE 0.5488	FALSE FAKE RATE (FFR) = 0.114 HALF TOTAL ERROR RATE (HTER) = 0.214	
Avg_R_PSNR 77.9648	HALF TOTAL ERROR RATE (HTER) = 0.214 fx >>	
Avg_R_RAMD 0.2295		
Ava R SME 199.7606		,

Figure 73: Fine KNN classification result of 1000 fingerprint images using 5 IQMs.

Name =   abdu   Abdu   Abdu   Abdu    Abdu   Abdu   Abdu    Abdu   Abdu   Abdu   Abdu   Abdu   Abdu   Abdu   Abdu   Abdu   Abdu   Abdu   Abdu    Abdu   Abdu   Abdu   Abdu    Abdu   Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu    Abdu   Abdu   A	📣 MATLAB R2016a										-	o ×
Have hash peage Curve Films Other Bins (Dethicition   Page Age Accession      Accession Conver Films	HOME PLOTS	APPS EDITO	IR PUE	BLISH VIEW					1196	🔁 🕐 Search De	ocumentation	ت <mark>م</mark>
<pre>     C v Uses &gt; LENOVO-PC &gt; Desktp &gt; hp. Newfolder (2) + Abdu code &gt;</pre>	Get More Install Package Curve		PID Tuner	System Signal Analysis	Image Instrument	-	C	Application Cla	assification	Distribution		
Immer Folder C   Imme 4   abdu   finger database   hiffest   3ANUArn   badum   abdu   abdum   badum   abdum   abdum   abdum   abdum   abdum   abdum   abdum   abdum   badum   abdum   abdum   abdum   abdum   abdum   abdum   badum   abdum   badum   badum   abdum   badum   badum   abdum   badum   badum   afterdencem   Aversgülffrencem   bestjoktm   bestjoktm   bestjoktm   bestjoktm   bestjoktm   vilsp('6-gradient phase error   bestjoktm   bestjoktm   bestjoktm   bestjoktm   bestjoktm   bestjoktm   bestjoktm   vilsp('6-gradient phase error = ');   signal to noise ratio   signal to noi	FILE				APPS							
Name *       Multiple in the second sec	🜩 🔁 🎏 📙 🕨 C: 🕨 Users 🕨	LENOVO-PC + Deskto	p ► hp_New fol	lder (2) 🕨 Abdu code 🕨								<b>-</b> .
abdu     V     Variation Magnitude Erfor       finger database     Value     Value       Mittest     Value     Value       Badudm     Value     Value       Badudm     Value     Value       Badudm     Value     Value       AbduSwedByTabits     Value     Value       AbdusKedByTabits     Value     Value       AbdusKedByTabits     Value     Value       Bett/Autom     Value     Value       Value     Value     Value       AversgdTerror     Value     Value       AversgdTeroror     Value <td>Current Folder</td> <td></td> <td>🕤 📝 Edi</td> <td>itor - C:\Users\LENOVO-PC\D</td> <td>esktop\hp_New folder (2)\Ab</td> <td>du code\best5_</td> <td>classification.m</td> <td></td> <td></td> <td></td> <td></td> <td></td>	Current Folder		🕤 📝 Edi	itor - C:\Users\LENOVO-PC\D	esktop\hp_New folder (2)\Ab	du code\best5_	classification.m					
tails	Body     Solutions		<ul> <li>77</li> <li>78</li> <li>79</li> <li>80</li> <li>81</li> <li>82</li> <li>83</li> <li>84</li> <li>85</li> <li>86</li> <li>87</li> <li>89</li> </ul>	<pre>style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-style="text-s</pre>	<pre>rtude erior nt magnitude error error haseError(origImg, nt phase error = ') e ratio oiseRatio(origImg,</pre>	<pre>= '); distImg); ; distImg);</pre>	ification.m ×	estŚwrk.m 🗙 [	Best_10.m	K bbbjhdel.m X	[+]	•
orkspace     O       ame A     Value       ans     1       Ang_F, OFE     NaM       Ang_F, SNE     NaM </td <td>Details</td> <td></td> <td>^ <b>_</b></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>,</td>	Details		^ <b>_</b>									,
ans     1       Avg_F_OPE     Number of Real images predict as Real = 453       Avg_F_PSN     Number of Real images predict as Fake = 47       Number of Real images predict as Fake = 47       Number of Real images predict as Fake = 104       Avg_F_SNR     Number of Fake images predict as Fake = 124       Avg_F_SNR     Number of Fake images predict as Real = 124       Avg_F_SNR     Number of Fake images predict as Real = 124       Avg_F_SNR     Number of Fake images predict as Real = 124       Avg_F_SNR     Number of Fake images predict as Real = 124       Avg_F_SNR     Number of Fake images predict as Real = 124       FALSE CENUINE RATE(FGR) = 0.044     HALF TOTAL ERROR BATE (HTER) = 0.171       Avg_F_SNR     0.235       Avg S.KME     19,7056	Workspace											
	ans         1           Avg.F_CPE         NaN           Avg.F_SNR         NaN           Avg.F_RAMD         NaN           Avg.F_SMR         NaN           Avg.F_SMR         NaN           Avg.F_SMR         NaN           Avg.F_SMR         NaN           Avg.F_SMR         NaN           Avg.R_CPE         0.5488           Avg.R_SNR         7.75648           Avg.R_AMD         0.2255		A Nu Nu Nu FA FA S	mber of Real images mber of Real images mber of Fake images LSE GENUINE RATE(FG LSE FAKE RATE(FFR): LF TOTAL ERROR RATE	predict as Real = predict as Fake = predict as Fake = predict as Real = R) = 0.248 = 0.094	47 376						
	Ava R SME 199.7606		Y (									>

Figure 74: Logistic Regression classification result of 1000 fingerprint images using 5 IQMs.

# 7. Table and Screenshot of Training Results For 100 Fingerprint Images Using 25 IQMs:

100 Fingerprint Images With 25 IQMs Result Using 6 Classifiers:

## 1. LDA

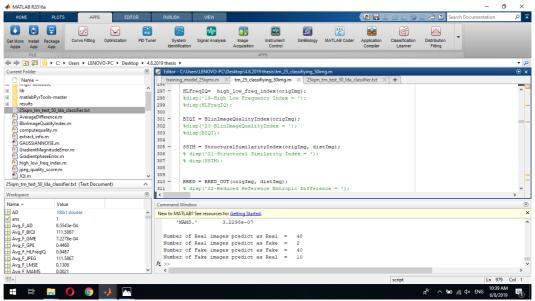


Figure 75: LDA training result of 100 fingerprint images using 25 IQMs.

HOME	PLOTS APPS	EDITO	R	PUBLISH	VEW					88.		280	Search Documenta	tion 🔎
t More Install Pack	age Curve Fitting	Optimization	PID Tuner	System Identificatio	Signal Analysis	Image Acquisition	Instrument Control	SimBiology	MATLAB Coder	Application Compiler	Classification Learner	Distribution Fitting	•	
FILE							APPS							
	C: • Users • LENO	VO-PC 🕨 Deskto												
rrent Folder			💿 🍃	🚪 Editor - C:\U	lsers\LENOVO-PC\D	esktop\4.8.20	19 thesis\tm_25_	classifiying_50i	mg.m					0
📄 Name 🔺				fraining_m	odel_25iqms.m 🛛 🛛	tm_25_clas	sifiying_50img.m	🛛 🛛 🕺 25iqm,	tm_test_50_lda_c	lassifier.txt 🔿	( <b>) +</b> ]			
lib			A 3	314 - JF	EG = jpeg_qua	lity_scor	e(origImg);							
matlabPyrTools	master		3	815 %	disp('23-JPEG	Quality	Score = ');							
results			1	816 %	disp(JPEG);									
25iqm_tm_test_	50_lda_classifier.txt			317										
🔄 AverageDifferen				818										
🔄 BlinlmageQualit					F = vifvec(or									
computequality	.m				disp('24- Vis	ual Infor	mation Fide	lity = ');						
extract_info.m					disp(VIF);									
GAUSSIANNOIS				322										
GradientphaseE					QE = NaturalI disp('25- Nat									
high low freq i					disp(NIQE);	uralimage	Qualityrvai	uator = ·)	÷					
jpeg_quality_sco				326	disp(arge);									
🕙 JQI.m				327										
m_tm_test_50_lda	classifier.txt (Text Docu	iment)			ector of fake	image fe	atures							
					(:,x)=[MSE; S			NR; PSNR;	NK; AD; SC;	MD; RAMD	NAE; TED;	TCD; LMSE	; MAS; MAMS;	HLFregIO;
rkspace				<ul> <li>.*</li> </ul>		1.1								
me 🔺	Value													
AD.	100x1 double		<u>^</u>	Command Win	dow									
ans	1			New to MATLA	B? See resources for	Getting Starte	ed.							
Avg_F_AD Avg_F_BIQI	8.5543e-04 111.5867													
Avg_F_BIQI Ava F GME	7.2278e-04			Number o	f Real images	predict	as Real =	49						
Avg_F_GPE	0.4468			Number o	f Real images	predict	as Fake =	1						
Avg_F_HLFregIQ	0.9487			Number o	f Fake images	predict	as Fake =	50						
Avg_F_JPEG	111.5867				f Fake images	predict	as Real =	0						
Avg_F_LMSE	0.1308			\$ >>										
Ava F MAMS	0.0021		*	<										
											script		1	Ln 438 Col

Figure 76: QDA Training Result of 100 Fingerprint Images Using 25 IQMs.

📣 MATLAB R2016a		– Ø ×
HOME PLOTS APPS EDITO	R PUBLISH VEW	🔄 🏦 🛱 😒 🖻 🕐 Search Documentation 🛛 🔎 💈
Get More Instal Package Apps App App	Image         Image <th< th=""><th>Classification Distribution Learner Fiting</th></th<>	Classification Distribution Learner Fiting
FILE	APPS	
🔶 🛅 🔀 📙 🕨 C: 🕨 Users 🕨 LENOVO-PC 🕨 Deskto		کر 👻
Current Folder	Editor - C:\Users\LENOVO-PC\Desktop\4.8.2019 thesis\tm_25_classifying_50img.m	⊙ >
Name A     Toget database     Ingred data	<pre>65 66 67 47Peak Signal to Noise Ratio 68 70 71 48Normalized Cross-Correlation 72 74 75 48Average Difference 76 70 71 78 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7</pre>	TM_Create_25m × +
	SIG Structural content	
Workspace	80 - SC = StructuralContent(origImg, distImg);	×
Name ∧         Value           AD         100/1 double           Ang F_AD         8,5543e-04           Ang F_BCQ         111,5867           Ang F_GOM         7,2278e-04           Ang F_GOM         0,4468           Ang F_FILTerQU         0,9487           Ang F_FILTERQU         0,9487           Ang F_LHEreqU         0,3103           Ang F_LHERO         0,1308           Ang F_LMSE         0,1308	Command Window     New to MATLAS? See resources for <u>Gatting Stated</u> Number of Real images predict as Real = 50     Number of Real images predict as Fake = 0     Number of Real images predict as Fake = 49     Number of Fake images predict as Real = 1     A >> of Fake images predict as Real = 1	Activate Windows Go to Settings to activate Windows
AVG P MAMS 0.0021		script In 78 Col 1

Figure 77: Linear SVM training result of 100 fingerprint images using 25 IQMs.

# 4. Quadaratic SVM

HOME PLOTS	APPS	EDITOR	PUBLIS	H VEW				a 8 2	614	@ 🗗 🕐 Se	arch Documentation	< 10 م
th More Install Package Curv	Fitting Optimiz	]	uner Sy	stem Signal Analysis	Image Instrumer Acquisition Control	t SimBiology	MATLAB Coder	Application Compiler	Classification Learner	Distribution Fitting	•	
FILE	LENOVO-PC	Derkton b	1 9 2010 theorie		APPS							
irrent Eolder	- EENOVO-FC	· Desktop ·	_		esktop\4.8.2019 thesis\tm	25 classificing 50	ima m					
Name A				na model 25iams.m				(1000).m ×	TM Create 25	m ×]+]		
Finger database Ibb mattabPyTcolo-master results 25igm, m. test, 30, ida, classifik AverageDifference m 6 BiningaQualityinder, m 6 Computequalityinde 6 Computequalityinde 7 Computequalityin	.bxt			%8Normalized Cr NK = Normalized %9Average Diffe AD = AverageDif %10_Structural	altoNoiseRatio (or oss-Correlation CrossCorrelation ( rence ference (origImg,	origImg, dis iistImg);	-					
orkspace		0	264									
ame 🔺 Value			<									>
AD 100x1 doubl ans 1 Avg_F_AD 8.5543e-04 Avg_F_BIQ1 11.5867 Avg_F_GRE 7.2278e-04 Avg_F_GRE 0.4468 Avg_F_HLFreqU 0.9487 Avg_F_PLF6 111.5867 Avg_F_LMSE 0.1308 Avg_F MMS 0.0021			Numbe Numbe Numbe	IATLAB? See resources for er of Real images er of Real images er of Fake images	Getting Started. predict as Real predict as Fake predict as Fake predict as Real	- 0 - 50				Activate V	/indows s to activate Windo	DWC

Figure 78: Quadaratic SVM training of 100 fingerprint images using 25 IQMs.

📣 MATLAB R2016a														- 0 :
номе	PLOTS APP	S EDITO	DR	PUBLISH	VIEW						4 B B 9	2 🖸 🕐	Search Docume	ntation 🔎
Get More Apps App Act		Optimization	PID Tuner	System Identificatio	Signal Analysis	Image Acquisition	Instrument Control	SimBiology	MATLAB Coder	Application Compiler	Classification Learner	Distribution Fitting	-	
FILE							APPS							
	C: Visers Visers	VO-PC • Deskt	·											-
Current Folder			ء 🕞	🚪 Editor - C:\U	Jsers\LENOVO-PC\[	Desktop\4.8.20	19 thesis\tm_25_	classifiying_50	img.m					C
📄 Name 🔺					iodel_25iqms.m 🛛 🏾	tm_25_clas	sifiying_50img.n	n 🛛 Main.i	m × TEST_15	_(1000).m 🛛 🖂	TM_Create_2	5.m × +		
🗉 📙 finger database				810 - RF	RED - RRED OUT	-	diam Terry) .							
lib					disp('22-Redu									
matlabPyrTools	-master				disp(RRED);	iced verer	ence Encro	ore priner	ence = ·);					
results				313	arss (mass),									
25iqm_tm_test_ AverageDifferent	50_lda_classifier.txt				PEG = jpeg qua	lity scor	e (origīma)							
BlinImageQualit					disp('23-JPE									
Computequality			3		disp(JPEG);									
extract info.m			3	817										
GAUSSIANNOIS	E.m		3	818										
🖄 GradientMagnit			3	19 - VI	IF = vifvec(or	rigImg, di	stImg);							
GradientphaseE			3	820 🐐	disp('24- Vis	sual Infor	mation Fide	<pre>ity = ')</pre>	;					
A high_low_freq_i					disp(VIF);									
jpeg_quality_sco	ore.m			322										
b (Folder)					QE = Natural:									
					disp('25- Nat	turalImage	QualityEval	uator = '	);					
Vorkspace	1				disp(NIQE);									
Name 🔺	Value			<										,
AD	100x1 double		^ <u>(</u>	Command Win	dow									
ans	1			New to MATLA	B? See resources fo	r Getting Starte	ed.							
Avg_F_AD	8.5543e-04 111.5867													
Avg_F_BIQI Avg_F_GME	7.2278e-04			Number o	f Real images	predict :	as Real =	50						
Avg_F_GPE	0.4468			Number o	f Real images	predict a	as Fake =	0						
Avg F HLFregIQ	0.9487			Number o	f Fake images	predict a	as Fake =	50						
Avg_F_JPEG	111.5867				f Fake images	predict a	as Real =	0				Activate	Windows	
Avg_F_LMSE	0.1308		f.	\$ >>									gs to activat	
Ava F MAMS	0.0021		×	<										c windows.
											script			Ln 977 Col

Figure 79: Fine KNN training result of 100 fingerprint images using 25 IQMs.

MATLAB R2016a HOME PLOTS APPS EDITOR	PU	JBLISH VIEW					1 6 5	@ E ? Se	- arch Documentation	< ات م
	PID Tuner	System Signal Analysis Identification	Image Instrument Acquisition Control	SimBiology	MATLAB Coder	Application Compiler	Classification Learner	Distribution	•	
FILE			APPS							
🖙 🔁 🔚 🕨 C: 🕨 Users 🕨 LENOVO-PC 🕨 Desktop	<ul> <li>4.8.2019 th</li> </ul>	nesis 🕨								-
rrent Folder	🐨 📝 Ed	ditor - C:\Users\LENOVO-PC\D	esktop\4.8.2019 thesis\tm_25_c	lassifiying_50ir	mg.m					C
Name 🛧	i t	training model 25igms.m 🛛 🗙	tm 25 classifiving 50img.m	× Main.m	n × TEST 15	(1000).m ×	TM Create 25	m ×l+l		_
finger database	A 200									
lib	269									
matlabPyrTools-master	270	<pre>%12_RaverageMD</pre>								
results	271	<ul> <li>RAMD = Rarverage</li> </ul>	eMD(origImg, distImg	);						
25iqm_tm_test_50_lda_classifier.txt	272									
AverageDifference.m	273									
BlinImageQualityIndex.m	274	%13_Normalized a								
computequality.m	275	<ul> <li>NAE = Normalize</li> </ul>	dAbsoluteError(origI	ng, distIm	ıg);					
extract_info.m	276									
GAUSSIANNOISE.m	277									
GradientMagnitudeError.m	278	<pre>%14_Total Edge  </pre>								
GradientphaseError.m	279	<ul> <li>TED = TotalEdge</li> </ul>	Difference(origImg,	distImg);						
high_low_freq_index.m	280									
jpeg_quality_score.m	<ul><li>281</li></ul>	%15_TotalCorner								
(Folder)	A 282	<ul> <li>TCD = TotalCorn</li> </ul>	erDifference(origImg	, distImg)	2					
	283									
kspace	284	<ul> <li>LMSE= Laplacia;</li> </ul>	nMeanSquaredError(or	igImg, dis	stImg);					
e 🔺 Value	<									>
D 100x1 double	Comr	mand Window								
ns 1	New	to MATLAB? See resources for	Gatting Started							
vg_F_AD 8.5543e-04	INCIV	to minimumorial ace resources for	Second Second							_
vg_F_BIQI 111.5867										
vg_F_GME 7.2278e-04		umber of Real images		50						
Avg_F_GPE 0.4468		umber of Real images		0						
ivg_F_HLFreqIQ 0.9487		umber of Fake images		50						
lvg_F_JPEG 111.5867		umber of Fake images	predict as Real =	0				Activate W	indows	
Avg_F_LMSE 0.1308	fx >>			_						
Ava F MAMS 0.0021	✓ <									

Figure 80: Logistic Regression training result of 100 fingerprint images using 25 IQMs.

## 8. Screenshot of Classification Results For 1000 Fingerprint Images

## Using All 25 IQMs

1000 Fingerprint images with 25 IQMs result using 6 classifiers:

#### 1. LDA

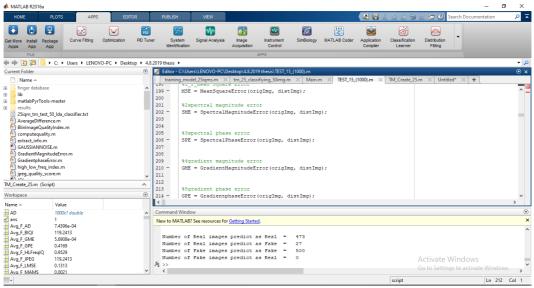


Figure 81: LDA classification result of 1000 fingerprint images using 25 IQMs.

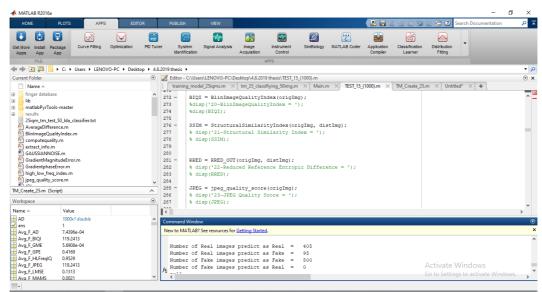


Figure 82: QDA classification result of 1000 fingerprint images using 25 IQMs.

📣 MATLAB R2016a		- 0 ×
HOME PLOTS APPS EDITOR Constraint Package Curve Fitting Optimization F	PUBLICH         VEW         Image         Image <th< th=""><th>Learner Ptting</th></th<>	Learner Ptting
FILE	APPS	را <del> -</del>
🛑 🜩 💽 🔀 📙 🕨 C: 🕨 Users 🕨 LENOVO-PC 🕨 Desktop Current Folder		•
Current Police           Imme ^           Imme ^		TM_Create_25m × +
Workspace	RAMD = RarverageMD(origImg, distImg); 243	
Name 🗠 Value	<	>
AD         1000:1 double           ans:         1           avg.f.B(0)         119.2413           avg.f.B(0)         119.2413           avg.f.B(0)         5.6908-04           avg.f.F.H(req)(0)         05539           avg.f.L(req)(0)         119.2413           avg.f.L(req)(0)         119.2413           avg.f.L(req)(0)         05393           avg.f.L(req)(0)         119.2413           avg.f.L(NSE)         0.1313           avg.f.L(NSE)         0.0021	Command Window New to MATLAB? Sec resources for <u>Getting Started</u> Number of Real images predict as Real = 487 Number of Real images predict as Pake = 13 Number of Fake images predict as Pake = 500 Number of Fake images predict as Real = 0	Activate Windows Go to Settings to activate Windows.
AVG F MAMS 0.0021		script In 242 Col 3

Figure 83: Linear SVM classification result of 1000 fingerprint images using 25 IQMs.

# 4. Quadaratic SVM

MATLAB R20163 HOME PLOTS APPS EDITOR Company PLOTS APPS EDITOR (Company Planta) Company Planta Package Curve Filling Optimization		- C >
Apps App App FILE	Identification Acquisition Control Compiler Les	amer Fitting
+ + T T + C: + Users + LENOVO-PC + Desktop		-
Current Eolder	Editor - C:\Users\LENOVO-PC\Desktop\4.8.2019 thesis\TEST_15_(1000).m	۲
Name - Na	<pre>A 275 A 275 A 275 A 275 A 275 A 275 A 276 A 351M - StructuralSimilarityIndex(origImg, distImg); A 276 A 351M - StructuralSimilarityIndex(origImg, distImg); A 37 /pre>	eate_25m × Untitled* × +
Workspace	290 - VIF = vifvec(origImg, distImg); 291 % disp('24- Visual Information Fidelity = ');	
Name 🔺 Value	291 % disp('24- Visual Information Fidelity = ');	>
AD 1000x1 double ans 1 Avg_F,AD 7.4396e-04 Avg_F,BIQI 119.2413	Command Window New to MATLAB? See resources for <u>Setting Started</u> . Number of Real images predict as Real = 497	
Avg_F_GME         5.6908e-04           Avg_F_GPE         0.4169           Avg_F_HIreqIQ         0.9529           Avg_F_HIRE         119.2413           Avg_F_LMMSE         0.1313           Avg_F EMMS         0.0021	Number 05 Real images predict as Fake = 3 Number of Fake images predict as Fake = 50 Number of Fake images predict as Real = 0 $f_{\rm R} > \sim$	Activate Windows Go to Settings to activate Windows.

Figure 84: Quadaratic SVM classification result of 1000 fingerprint images using 25 IQMs.

Image: Instant Package       Curve Fitting Optimization PD Tuner       System Syntal Analysis Package       Syntal Analysis Package       System Syntal Analysis Package       S	📣 MATLAB R2016a										- a ×
Bit     Curre Trim     Optimize     Optimize     System     Syst	HOME PLO	OTS APPS	EDITOR	PUBLI	SH VIEW					🛍 🗇 🗠 🖨 🕐 Sea	rch Documentation 👂
<pre>     C</pre>	Get More Install Packag			PID Tuner S	ystem Signal Analysis	Image Inst	rument SimBiology	C	Application Class	sification Distribution	-
ament Folder       C       Editor - CAUseryLINUVC-PCODesition 44.32019 thesin/LES1 [15,(1000).m       C         I ham e A       Tinger database       Value       Value <td></td> <td></td> <td></td> <td></td> <td></td> <td>APPS</td> <td></td> <td></td> <td></td> <td></td> <td></td>						APPS					
Name +         Inger database		C: ▶ Users ▶ LENO\	VO-PC • Desktop								•
figs datase     est     Function       bit     ass     figs datase       mile     ass     figs datase       bit     ass     figs datase       computeguistion     ass     figs datase       datase     figs datase     figs datase       figs datase     figs datase     figs datase <t< td=""><td>urrent Folder</td><td></td><td></td><td>-</td><td></td><td></td><td></td><td></td><td></td><td></td><td>0</td></t<>	urrent Folder			-							0
Ortspace         O         273         tdisp(*20-81inTmageQualityIndex = ');         O           ame +         Value         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >         >	finger database iib mattabPyrTools-m results 2 Sigm_tm_test_50 A verageDifference B BinimageQuality/r computequality.m extract_info.m G GAUSSIANNOISE. G radientMagnitud G radientphaseError hing_low_free_indality_core	lda_classifier.txt .m ndex.m n eError.m r.m ex.m		<ul> <li>▲ 258</li> <li>259 -</li> <li>260</li> <li>261</li> <li>262</li> <li>263</li> <li>264 -</li> <li>265</li> <li>266</li> <li>267</li> <li>268 -</li> <li>269</li> <li>✓</li> <li>270</li> <li>271</li> </ul>	<pre>Wursg(think); MAS = MeanAngle %disp('17-Mean %disp(MAS); %Mean Angle Mag MAMS = MeanAngl %disp('18-Mean %disp(MAMS); HLFreqIQ= high %disp('19-High %disp('19-High %disp('19-High %disp('19-High %disp('19-High %disp('19-High %disp('19-High)</pre>	eSimilarity(ori Angle Similari gnitude Similari Angle MagnitudeSimi Angle Magnitude Low Frequency );	<pre>gImg, distImg); ty = '); ity larity(origImg, e Similarity = x(origImg); Index = ');</pre>	distImg);	1000).m 🗶 📶 🤇	reate.25.m × Untitled	* *   +
AD     1000x1 floable        AD     1000x1 floable        ans     1       Avg_F,AD     7.4386-04       Avg_F,GM     56908-04       Avg_F,GM     56908-04       Avg_F,GM     56908-04       Avg_F,GM     0189       Avg_F,BRQ     0189       Avg_F,BRQ     0189       Avg_F,BRQ     0189       Avg_F,BRQ     0189       Avg_F,BRQ     0189       Avg_F,BRQ     01313       Avg F,MRS     0.0021       V     Go to Settings to activate Windows	Vorkspace										
AD     1000x1 double     A     Command Window       ans     1     New to MATLAB? See resources for Getting Stated.       Ang_f_B01     1192413     New to MATLAB? See resources for Getting Stated.       Ang_f_F0F     0.4166     Number of Real images predict as Real = 494       Number of Real images predict as Fake = 6     Number of Real images predict as Fake = 6       Number of Fake images predict as Fake = 500     Number of Fake images predict as Real = 0       Ang_f_M5K     0.0313       Ang_FMMS     0.0313		Value			salop( zo-biim	image gadil by inc					>
Ava F MAMS 0.0021	AD Ang_F_AD Avg_F_BIQI Avg_F_GME Avg_F_GME Avg_F_GME Avg_F_GPE Avg_F_ILfreqIQ Avg_F_JEG Avg_F_JEG Avg_F_JEG Avg_F_JEG	1000x1 double 1 7.4396e-04 119.2413 5.6908e-04 0.4169 0.9529 119.2413 0.1313		New to I Numi: Numi: Numi: Numi: Numi: Numi: Numi: Numi: Numi:	MATLAB? See resources for over of Real images over of Real images over of Fake images	s predict as Re s predict as Fa s predict as Fa	ke = 6 ke = 500				
	H Ava F MAMS (	10021							- coni	ot	Ln 267 Col

Figure 85: Fine KNN classification result of 1000 fingerprint images using 25 IQMs.

HOME PLOTS APPS	EDITOR PUE	ILISH VIEW				🖥 🗇 定 🔁 🕐 Search I	Documentation 🔎
More Instal Package pps App App	zation PID Tuner	System Signal Analysis Ientification	Image Instrument Acquisition Control	SimBiology MATLAB Cod	er Application Classi	fication Distribution arner Fitting	
FILE			APPS				
🔶 🛅 🔀 📙 🕨 C: 🕨 Users 🕨 LENOVO-PC							•
rent Folder	🕤 🗾 Edi	tor - C:\Users\LENOVO-PC\D	esktop\4.8.2019 thesis\TEST_15_(	1000).m			0
Name 🔺	🗐 🗍 tra	aining_model_25iqms.m 🛛 🛛	tm_25_classifiying_50img.m	× Main.m × TEST_	15_(1000).m 🛛 🛛 TM_Cr	eate_25.m × +	
finger database	112 -		Similarity(origImg, d:				
lib	113		Angle Similarity = '	4			
matlabPyrTools-master	114	%disp(MAS);					
25igm tm test 50 lda classifier.txt	115						
AverageDifference.m	116		nitude Similarity				
BlinImageQualityIndex.m	117 -		eMagnitudeSimilarity(				
computequality.m	118		Angle Magnitude Simila	rity = ');			
extract_info.m	119	%disp(MAMS);					
GAUSSIANNOISE.m	120	UT Francisco bish	low freq index(origIn				
GradientMagnitudeError.m	121 -		Low Frequency Index =				
GradientphaseError.m	123	%disp(HLFreqIQ)		11			
high_low_freq_index.m	124	(introduct)					
jpeg_quality_score.m	× 125 -	BIOI = BlinImag	eQualityIndex(origImg)				
Folder)	^ 126		mageQualityIndex = ')				
kspace	(v) 127	%disp(BIQI);					
	128						
ne 🔶 Value	<						>
D 1000x1 double	^ Comm	and Window					
14	New t	o MATLAB? See resources for	Getting Started				
vg_F_AD 7.4396e-04 vg F_BIQI 119.2413							
vg_F_6IQI 119:2415	Nu	mber of Real images	predict as Real =	495			
vg_F_GPE 0.4169		mber of Real images		5			
vg_F_HLFreqIQ 0.9529		mber of Fake images		500			
vg_F_JPEG 119.2413		mber of Fake images	predict as Real =	0		Activate Wind	
Avg_F_LMSE 0.1313	fx >>						
Ava F MAMS 0.0021	Y <						

Figure 86: Logistic Regression classification result of 1000 fingerprint images using 25 IQMs.