Parametric Real Face Images Detection System (RFIDS) Using Multiple Classifiers

Mohammed Osman Mohammed

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 2017
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Mustafa Tümer
Director

I certify that this thesis satisfies 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. Gurcu Öz
3. Assoc. Prof. Dr. Önsen Toygar

ABSTRACT

Recently biometric researches against spoofing attacks has been an important role of study, today we can examine the improvement of this biometric security technology against challenging methods such as spoofing attacks.

In this thesis software-based approach is presented based on image quality assessments (IQA) to discriminate real genuine face images from impostor samples, a liveness assessment method is added to the present system to ensure friendly use, processing speed, and non-intrusive biometric system.

The proposed method RFIDS uses 15 image quality features to decrease the level of complexity and make the system applicable for real-time applications. The experimental results achieved from this implemented work on an available dataset generates a high degree of positive detection compared to other existing methods and that the 15 image quality measures (parameters) are efficient in classifying real faces from printed impostor samples. There are some useful information retrieved from real images using IQA that makes the system capable enough to discriminate them from printed traits.

Keywords: Image quality assessment, biometric, real and spoof face detection.

ÖZ

Gerçek ve sahte yüz görüntüleri arasında ayrım yapmak, biyometrik kimlik doğrulama araştırmalarında önemli bir yer tutmuştur ve son zamanlarda biyometrik sistemlerde koruma geliştirmek için bu alan üzerinde araştırmalar yapılmıştır.

Bu tezde, yazılım tabanlı yaklaşım olarak Görüntü Kalitesi Değerlendirme (IQA) yöntemteri kullanilmistir Gerçek orijinal yüz imgelerini sahte örneklerden ayırabilmek için, kolay kullanım, işleme hızı ve müdahaleci olmayan biyometrik sistemi sağlamak için mevcut sisteme bir “canlılık değerlendirme yöntemi” eklenmiştir.

Önerilen yöntem, karmaşıklık seviyesini azaltmak ve sistemi gerçek zamanlı uygulamalar için uygun hale getirmek için 15 görüntü kalitesi özelliğini kullanmaktadır. Literatürde kullanılan bir veri kümesi üzerinde uygulanan bu çalışmadan elde edilen deneysel sonuçlar, diğer mevcut yöntemlere kıyasla yüksek derecede pozitif algılama üretir ve 15 görüntü kalitesi ölçütü, basılı sahte örneklerden gerçek yüzleri sınıflandırmada verimli olur. IQA kullanarak gerçek görüntülerden elde edilen bazı bilgiler, onları, basılı görüntülerden ayırt edebilecek kadar sistemi yeterli kılan bir yapıya getirmiştir.

Anahtar Kelimeler: Görüntü kalitesi değerlendirmesi, biyometri, gerçek ve sahte yüz saptama.

DEDICATION

This thesis work is dedicated to my parents, who have been a constant source of support and encouragement during the challenges of graduate school and life. I am truly thankful for having you in my life, who have always loved me unconditionally and whose good examples have taught me to work hard for the things that I aspire to achieve. I am grateful.

To my brothers and sisters, to the long nights you spent helping me getting through this thesis, the long hours you spent encouraging me that it will all be over, to the times when it was impossible to continue, you were all standing by my side with your encouraging words and love, this work would not have been complete with all of you. I am lucky to have you all in my life. Thank you.

Not least of all, I owe so much to my friends for their undying support, Unfortunately, I cannot thank everyone by name, I just want you all to know that you count so much. Had it not been for all your prayers and benedictions; were it not for your sincere love and help, I would never have completed this thesis. So thank you all.

ACKNOWLEDGMENT

In the Name of Allah, the Most Merciful, the Most Compassionate all praise be to Allah, the Lord of the worlds; and prayers and peace be upon Mohamed His servant and messenger.

First and foremost, I must acknowledge my limitless thanks to Allah, the Ever Magnificent; the Ever-Thankful, for His helps and bless. I am totally sure that this work would have never become truth, without His guidance.

My sincere appreciation also goes to my supervisor Assoc. Prof. Dr. Alexander Chefranov, whose contribution and constructive criticism has pushed me to expend the kind of efforts I have exerted to make this work as original as it can be. Thanks to him I have experienced true research and my knowledge on the subject matter has been broadened. I will never forget you sir.

Special thanks also go to Assoc. Prof. Dr. Önsen Toygar on her comments on the study, and valuable advice.

A special thanks to my friends, for helping me through the process of finalizing this thesis, no words can express my gratitude to your efforts, thanks for your assistance and your kind words, you will be in my mind.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AD	Average Difference
ANOVA	Analysis Of Variance and Regression Analysis
ANN	Artificial Neural Network
APCER	Attack Presentation Classification Error Rate
BPCER	Bona Fide Presentation Classification Error Rate
BIQI	Blind Image Quality Index Measurement
CQ	Correlation Quality
DPA	Discriminant Power Analysis
EER	Equal Error Rate
ERMSC	Error Mean Squared Contrast
FAR	False Accept Rate
FFR	False Fake Rate
FGR	False Genuine Rate
FPR	False Positive Rate
FNR	False Negative Rate
FR-IQA	Full Reference Image Quality Assessment
FR	Full Reference
GME	Gradient Magnitude Error
GPE	Gradient Phase Error
GT	GrandTest
HTER	Half Total Error Rate
HLFI	High-Low Frequency Index
IF	Image Fidelity

JQI	JPEG Quality Index
LDA	Linear Discriminant Analysis
LINEAR-SVM	Linear Support Vector Machine
LMSE	Laplacian MSE
LIB-SVM	Library Support Vector Machine
MAS	Mean angle Similarity
MAMS	Mean angle Magnitude Similarity
MD	Maximum Difference
MSE	Mean Squared Error
MP	Matte Screen Photo
MV	Matte Screen Photo
NAE	Normalized Absolute Error
NCC	Normalized Cross Correlation
NR-IQA	No Reference Image Quality Assessment
NIQE	Natural Image Quality Evaluator
NR	No Reference
PDA	Personal Digital Assistance
PH	Print-Hand
PF	Print-Face
PCA	Principle Component Analysis
PSNR	Peak Signal To Noise Ratio
PMSE	Peak Mean Squared Error
QDA	Quadratic Discriminant Analysis
QUAD-SVM	Quadratic Support Vector Machine
RAMD	R-Averaged MD

RRED	Reduced Reference Entropy Difference
RBF	Radial Basis Function
RMSE	Root Mean Squared Error
RFIDS	Real Face Image Detection System
SC	Structural Content
SVM	Support Vector Machine
SME	Spectral Magnitude Error
SPE	Spectral Phase Error
SSIM	Structural Similarity Index Measurement
SSEQ	Spatial Spectral Entropy Quality
SNR	Signal to Noise Ratio
TCD	Total Corner Difference
TED	Total Edge Difference
TNR	True Negative Rate
VIF	Visual Information Fidelity

Chapter 1

INTRODUCTION

Nowadays, Biometric Recognition, or Biometrics can be defined as the recognition of individuals based on their physical and/or behavioral characteristics, is a prominent field of research [1]. Although among all the biometrics like: face, fingerprint, iris, signature etc. face has an outstanding importance over other systems because it's reliable, cheap and non-intrusive [2]. Although it's affected by some changes in sunglasses, lighting, facial hair etc. but all these affections can be enhanced using some filtering process.

There are different threats that detect such systems such as spoofing attacks which has been an important and motivated area for biometric researchers to investigate on such types of actions in areas such as iris [3], fingerprint [4], face [2], etc…

In such spoofing-attacks hackers use some synthetically produced materials such as gummy finger, printed faces or iris images, or try to copy the behavior of the genuine user such as signature [5], to access the system. Since these attacks are performed in the analogue domain with regular identifications, the usual known protection mechanisms are not effective such as encryption, watermarking or digital signature.

The number of different works done on this particular field has shown the necessity of implementing an advanced protection strategy to ensure more security [1]. Researchers in the recent years have focused on finding some specific quality measurements that changes the modification of biometric systems in order to target impostor samples and reject them, using this strategy to increase the security level of the biometric system.

This quality assessment method must be developed to ensure and satisfy some important needs [6]:

1) Non-intrusive: the proposed work should not have any degree of harmful contact with the user.
2) User friendly: users should not hesitate using the system.
3) Processing time: results should be taken out in a short interval for users not be connected for a large amount of time with the biometric sensor
4) Price: the cost should be affordable to increase the amount of users.
5) Performance: the system should have a low percentage of false fake rate (FFR) which indicates the real samples identified incorrectly as fake and false genuine rate (FGR) which indicates the fake samples identified incorrectly as real, for users confident when interacting with the system.

The system can be divided into four stages:

a) Image acquisition from user.
b) Apply Gaussian filter to image.
c) Calculate image quality measures (feature extraction).
d) Classification to discriminate between genuine and impostor samples.

Liveness detection methods can be classified into 2 approaches [6]:

1) Hardware-based approach: A specific machine is added to the sensor in a biometric system in order to measure some properties such as sweat, or facial hair etc.

2) Software-based approach: A system where an impostor users is recognized once their biometric traits are acquired using a normal sensor.

Somehow these two methods have benefits and downsides, which means a combination of both can give a superior protection approach to develop security of biometrics systems[7][8].

In the thesis, we implement a real face image detection software system (RFIDS) using image quality assessment (IQA), with different classifiers to ensure the quality of our system that gives a good level of real face image detection. The rest of the thesis is organized as follows. Section 2 presents a brief literature survey of existing methods based on spoofing detection and the problem definition, Section 3 is our implementation which presents a general diagram of our system and consist of how we implemented feature extraction and classifiers and we used Gaussian distribution to examine the feature implemented, Section 4 contains description of our experiments done on RFIDS and it shows the experimental setup and results obtained, Section 5 concludes our work and discusses the future work.

Chapter 2

SURVEY OF EXISTING RFIDS AND PROBLEM DEFINITION

2.1 Structure of IQA Method

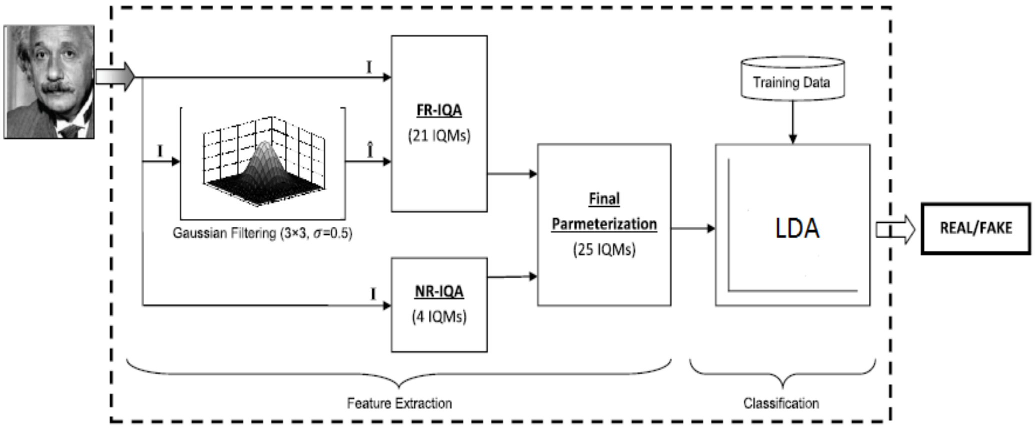


Figure 1: Structure of IQA Method [1]

IQM stands for image quality measurement, FR is full reference, and NR is no reference. The input image will be filtered using a Gaussian filter for calculating FR-IQA and the NR-IQA only operates with original image, at the final step of feature extraction the 25 IQA-measures (parameters) are combined, a classification method is applied to classify real or fake samples.

Steps of IQA method:

1) First the training model has to be obtained for obtaining training model a number of input images of known users fake and real have to be trained using

LDA classification method for the system to classify in further stages according to this training model.

2) Input image for classification : a gray scale image will be input to the system for classification

3) Gaussian filter: a Gaussian filter with 3*3 kernel and σ = o.5 will be introduced to the image in order to obtain 2 images original and enhanced image using Gaussian filter

4) Feature extraction: 25 features will be calculated for the input image

5) The last step will be classification process where LDA method is introduced, the inputs to this stage are two: training model and input image and according to the training model and LDA classification method the image is classified as either fake or real.

2.2 Gaussian Filter

Gaussian filter or Gaussian blur [42] in image processing is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise and reduce detail. The visual effect of this blurring technique is a smooth blur resembling that of viewing the image through a translucent screen, distinctly different from the bokeh effect produced by an out-of-focus lens or the shadow of an object under usual illumination.

Mathematically, applying a Gaussian blur to an image is the same as convolving the image with a Gaussian function.

The Gaussian blur is a type of image-blurring filters that uses a Gaussian function (which also expresses the normal distribution in statistics) for calculating the transformation to apply to each pixel in the image. In two dimensions, it is the product of two such Gaussians, one in each dimension:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.24}$$

Equation (2.24) is provided in [44]

$$\tilde{I}(i,j) = \sum_{x=-\lceil 3\sigma \rceil}^{\lceil 3\sigma \rceil} \sum_{y=-\lceil 3\sigma \rceil}^{\lceil 3\sigma \rceil} I(i+x, j+y) G(x,y) \tag{2.25}$$

Equation (2.25) is provided in [44]

Where – [3 σ] <= x <= [3 σ] and – [3 σ] <= y <= [3 σ].

An image with Gaussian blur distortion is given by `I = I * G

X = the distance from the origin in the horizontal axis

Y= the distance from the origin in the vertical axis

σ = the standard deviation of Gaussian distribution

When applied in two dimensions [42], this formula produces a surface whose contours and concentric circles with Gaussian distribution from the center point. Values from this distribution are used to build a convolution matrix which is applied to the original image.

The implementation of Gaussian blur effect is typically generated by convolving an image with a kernel of Gaussian values. In practice, it is best to take advantage of the Gaussian blur's separable property by dividing the process into two passes. In the first pass, a one-dimensional kernel is used to blur the image in only the horizontal or vertical direction. In the second pass, the same one-dimensional kernel is used to blur in the remaining direction. The resulting effect is the same as convolving with a two-dimensional kernel in a single pass, but requires fewer calculations.

2.3 Definitions of Known Image Quality Measures and Classifiers

In [9], defines 26 image quality measures and two types of classification methods. The presented measures are divided into two parts, FR IQA that is referred to full reference image quality measures which extracts quality features using two images, input image and the enhanced version of the same image using Gaussian filter, and NR IQM that refers to no reference image quality measures, these features are used to evaluate the condition of the real sample. This method [9] extracts 26 IQA features to reduce the level of complexity. It uses a discriminant analysis to discriminate between real and fake images namely linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA).

The 26 image quality measures (parameters) in [9] are as follows:

Mean Squared Error (MSE) [10]

Peak Signal to Noise Ratio (PSNR) [11]

Signal to Noise Ratio (SNR) [12]

Structural Content (SC) [13]

Maximum Difference (MD) [13]

Average Difference (AD) [13]

Normalized Absolute Error (NAE) [13]

R-Averaged MD (RAMD) [10]

Laplacian MSE (LMSE) [13]

Normalized Cross Correlation (NCC) [13]

Mean angle Similarity (MAS) [10]

Mean angle Magnitude Similarity (MAMS) [10]

Total Edge Difference (TED) [14]

Total Corner Difference (TCD) [14]

Spectral Magnitude Error (SME) [15]

Spectral Phase Error (SPE) [15]

Gradient Magnitude Error (GME) [16]

Gradient Phase Error (GPE) [16]

Structural Similarity Index Measurement (SSIM) [17] [18]

Visual Information Fidelity (VIF) [19] [18]

Reduced Reference Entropy Difference (RRED) [20] [18]

JPEG Quality Index (JQI) [21] [18]

High-Low Frequency Index (HLFI) [22] [18]

Blind Image Quality Index Measurement (BIQI) [23] [18]

Natural Image Quality Evaluator (NIQE) [24] [18]

Spatial Spectral Entropy Quality (SSEQ) [25] [18]

In our next subsections, we give detailed explanations of these measures.

2.3.1 FR Image Quality Assessment Measures

Full reference measures are divided into five different parts [9], 11 pixel difference measures, 2 edge based measures, 2 spectral distance measures, 2 gradient based measures, and 3 information theoretic measures, explained below:

1) Pixel difference measures:

    1) Mean Squared Error (MSE): is a measure that estimates the sum of squared difference (Error) between the input and enhanced image.

    The equation is:

$$MSE(\mathbf{I}, \hat{\mathbf{I}}) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (\mathbf{I}_{i,j} - \hat{\mathbf{I}}_{i,j})^2 \qquad (2.1)$$

Equation (2.1) is provided in [10]

2) Peak Signal To Noise Ratio (PSNR): this term is used to measure the ratio between the signal power and distortion noise, the equation is:

$$PSNR(\mathbf{I}, \hat{\mathbf{I}}) = 10 \log\left(\frac{\max(\mathbf{I}^2)}{MSE(\mathbf{I}, \hat{\mathbf{I}})}\right) \qquad (2.2)$$

Equation (2.2) is provided in [11]

PSNR is used to measure the loss of quality when image is compressed, the real data in PSNR is assumed to be the signal, and the noise is the loss introduced when image is compressed, measured in

Decibel (DB).

3) Signal To Noise Ratio (SNR): this measure is used to contrast the useful signal level to the noise level introduced by the background,

SNR is known as the rate of power in the input signal to the rate of the noise power, it is also referred to the ratio of wanted information to unwanted. The equation is given by:

$$SNR(\mathbf{I}, \hat{\mathbf{I}}) = 10 \log\left(\frac{\sum_{i=1}^{N} \sum_{j=1}^{M} (\mathbf{I}_{i,j})^2}{N \cdot M \cdot MSE(\mathbf{I}, \hat{\mathbf{I}})}\right) \qquad (2.3)$$

Equation (2.3) is provided in [12]

4) Structural Content(SC): is characterized as the summation square of original input image divided by the summation of enhanced image squared, the formula is:

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

Equation (2.4) is provided in [13]

5) Maximum difference (MD): it is the absolute maximum difference between the original and enhanced image, the equation is:

$$MD(\mathbf{I}, \hat{\mathbf{I}}) = \max |\mathbf{I}_{i,j} - \hat{\mathbf{I}}_{i,j}| \qquad (2.5)$$

Equation (2.5) is provided in [13]

6) Average difference (AD): is known as the sum of difference between the original and distorted image averaged by the number of image pixels, the formula is as follows:

$$AD(\mathbf{I}, \hat{\mathbf{I}}) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (\mathbf{I}_{i,j} - \hat{\mathbf{I}}_{i,j}) \qquad (2.6)$$

Equation (2.6) is provided in [13]

7) Normalized Absolute Error (NAE): is the summation of absolute difference between original and enhanced image divided by the summation of the absolute original image, its equation is known as:

$$NAE(\mathbf{I}, \hat{\mathbf{I}}) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} |\mathbf{I}_{i,j} - \hat{\mathbf{I}}_{i,j}|}{\sum_{i=1}^{N} \sum_{j=1}^{M} |\mathbf{I}_{i,j}|} \qquad (2.7)$$

Equation (2.7) is provided in [13]

8) R-Averaged MD (RAMD): is known as maximum difference summation of R between the real and enhanced images averaged by R value, the equation is:

$$RAMD(\mathbf{I}, \hat{\mathbf{I}}, R) = \frac{1}{R} \sum_{r=1}^{R} \max_r |\mathbf{I}_{i,j} - \hat{\mathbf{I}}_{i,j}| \qquad (2.8)$$

Equation (2.8) is provided in [10]

Where maxr is known as the r highest pixel difference between our original and enhanced image. In the present implementation r=10.

9) Laplacian-MSE (LMSE): is known as the sum ratio between the difference of the original and distorted image to the original image squared

Where h(**I**i,j)= **I**i+1, j +**I**i−1, j +**I**i, j+1 + **I**i, j−1 − 4**I**i, the equation is given as:

$$LMSE(\mathbf{I},\hat{\mathbf{I}}) = \frac{\sum_{i=1}^{N-1}\sum_{j=2}^{M-1}(h(\mathbf{I}_{i,j})-h(\hat{\mathbf{I}}_{i,j}))^2}{\sum_{i=1}^{N-1}\sum_{j=2}^{M-1}h(\mathbf{I}_{i,j})^2} \quad (2.9)$$

Equation (2.9) is provided in [13]

10) Normalized Cross Correlation (NCC): it is a standard image processing equation used for adjusting the brightness and normalization, it is known as the rate of summation when multiplying the real and enhanced sample, divided by summation squared of the original image, NCC equation is as follows:

$$NCC(\mathbf{I},\hat{\mathbf{I}}) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}(\mathbf{I}_{i,j}\cdot\hat{\mathbf{I}}_{i,j})}{\sum_{i=1}^{N}\sum_{j=1}^{M}(\mathbf{I}_{i,j})^2} \quad (2.10)$$

Equation (2.10) is provided in [13]

11) Mean Angle Similarity (MAS): is known as the mean angle that measures the similarity of the original sample when compared with enhanced samples the formula is as follows:

$$MAS(\mathbf{I},\hat{\mathbf{I}}) = 1 - \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(\alpha_{i,j}) \quad (2.11)$$

Equation (2.11) is provided in [10]

12) Mean Angle Magnitude Similarity (MAMS): can be defined as the mean angle that measures the magnitude similarity of original when compared enhanced samples, the formula is:

$$MAMS(\mathbf{I},\hat{\mathbf{I}}) = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(1-[1-\alpha_{i,j}][1-\frac{||\mathbf{I}_{i,j}-\hat{\mathbf{I}}_{i,j}||}{255}]) \quad (2.12)$$

Equation (2.12) is provided in [10]

2) Edge Based Measures:

1) Total Edge Difference(TED): the absolute difference of edges between the original and distorted image averaged by the value of image pixels, its formula is as follows:

$$TED(\mathbf{I},\hat{\mathbf{I}}) = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}|\mathbf{I}_{\mathbf{E}i,j}-\hat{\mathbf{I}}_{\mathbf{E}i,j}| \quad (2.13)$$

Equation (2.13) is provided in [14]

2) Total Corner Difference (TCD): is known as absolute value of subtraction when summing the corners of original samples from distorted samples then averaged by the maximum image number of corners, it is given in the equation:

$$TCD(I,\hat{I}) = \frac{|N_{cr}-\hat{N}_{cr}|}{\max(N_{cr},\hat{N}_{cr})} \quad (2.14)$$

Equation (2.14) is provided in [14]

3) Spectral Distance Measures:

1) Spectral Phase Error (SPE): is defined as summation of Fourier angle in distorted sample subtracted from Fourier angle of real sample squared and divided by the summation of image pixel, the formula is:

$$SPE(\mathbf{I},\hat{\mathbf{I}}) = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}|\arg(\mathbf{F}_{i,j}) - \arg(\hat{\mathbf{F}}_{i,j})|^2 \quad (2.15)$$

Equation (2.15) is provided in [15]

2) Spectral Magnitude Error (SME): is defined as summation error introduced from difference between the absolute Fourier transform of original image and the absolute Fourier transform of the enhanced image squared and averaged by the total number of image pixels, the equation is:

$$SME(\mathbf{I},\hat{\mathbf{I}}) = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(|\mathbf{F}_{i,j}| - |\hat{\mathbf{F}}_{i,j}|)^2 \quad (2.16)$$

Equation (2.16) is provided in [15]

2D Fourier transform Equation:

$$F[k,l] = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f[m,n]e^{-j2\pi\left(\frac{k}{M}m+\frac{l}{N}n\right)} \quad (2.17)$$

Equation (2.17) is provided in [15]

3) Gradient Based Measures:

1) Gradient Phase Error (GPE): is known as summing the difference of the absolute gradient value angle in the original sample and angle value of absolute gradient in enhanced sample divided the summation of image pixels, the formula is:

$$GPE(\mathbf{I},\hat{\mathbf{I}}) = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}|\arg(\mathbf{G}_{i,j}) - \arg(\hat{\mathbf{G}}_{i,j})|^2 \quad (2.18)$$

Equation (2.18) is provided in [16]

2) Gradient Magnitude Error (GME): is known as the sum of difference between the absolute gradient of the original image and the absolute

difference of the enhanced image squared divided by the total number of image pixels, the equation is given as:

$$GME(\mathbf{I}, \hat{\mathbf{I}}) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (|\mathbf{G}_{i,j}| - |\hat{\mathbf{G}}_{i,j}|)^2 \tag{2.19}$$

Equation (2.19) is provided in [16]

3) Information Theoretic Measures:

1) Structural Similarity Index Measurement (SSIM): upgrade of Widespread index, can be defined as the quality measurement when a single image is contrasted, and the other image is with its original quality.

   (See [17] and practical implementation in [18])

2) Visual Information Fidelity (VIF): VIF assumes that real face images are on scenes described as natural and based on this they should have same types of properties.

   (See [19] and practical implementation in [18])

3) Reduced Reference Entropy Difference (RRED): this measurement process as using wavelet to extract some local information's of the given sample and some speculation of the sample is not visible in samples in nature.

   (See [20] and practical implementation in [18])

2.3.2 No Reference Image Quality Measures

a) Distortion specific measures:

1) High Low Frequency Index (HLFI): it's sympathetic with sharpness and works by estimating the power difference between low and up frequency actions of Fourier spectrum.

$$SME(\mathbf{I}) = \frac{\sum_{i=1}^{i_l}\sum_{j=1}^{j_l}|\mathbf{F}_{i,j}| - \sum_{i=i_h+1}^{N}\sum_{j=j_h+1}^{M}|\mathbf{F}_{i,j}|}{\sum_{i=1}^{N}\sum_{j=1}^{M}|\mathbf{F}_{i,j}|} \quad (2.20)$$

Equation (2.20) is provided in [22]

2) JPEG quality index (JQI): it evaluates image qualities distorted by known closed artificial initiated when comparing algorithms at a decreased number of bit rate as JPEG. (See [21] and practical implementation in [18])

b) Training based measures [9]:

1) Blind Image Quality Index Measurement (BIQI): This technique is known in the past to train images, the idea behind this mode is that clear real images introduce some regular properties if calculated properly, aberrance of the uniformity in statistics presented in nature is able to calculate the quality of the given image. (See [23] and practical implementation in [18]).

c) Natural scene statistic approaches:

1) Spatial Spectral Entropy Quality (SSEQ): this quality can be calculated by converting the input image to spatial and spectral format, using Fourier transform the entropy amounts are evaluated, then match the two entropy values, calculate and consider the inequality between them.
(See [25] and practical implementation in [18]).

2) Natural Image Quality Evaluator (NIQE): This measurement is known as the evaluation of blind image quality when extracting features of statistics associated to many alterations generating quality information's.
(See [24] and practical implementation in [18])

2.3.3 Classification Methods Results are Plotted in Terms of

1) Scatter Plot [35]: is also known as scatter graph or chart, the input in these chart is two variables, with the use of Cartesian coordinate these variables values are plotted and displayed. These values are displayed in a number of points, each point has a value representing one variable showing the position on horizontal axis, and value showing the position in vertical axis.
2) Confusion Matrix[36]: it is also known as error matrix, it is composed in machine learning field, it is a table that views the efficiency of an algorithm , each column in the matrix show the occurrence in a predicted class where the row shows the occurrence in the actual class.
3) ROC Curve [37]: it is a graphical plot that represents the achievement of a binary classification system where the classification threshold is assorted. True positive and false positive rates are uses in plotting the curve using an assorted threshold settings
4) Parallel Coordinates Plot [38]: it is used to visualize high dimensional geometry and to analyze data, it also represents a number of points in an n-dimension space, parallel lines are drawn in a vertical manner with equal spaces, the represented point in n-dimension space is a polyline with vertices sown on the parallel axes, the vertex position on the j-the axis correlates to the j-th coordinate of the point.

2.3.4 Classification of Real and Fake Face Images

This classification stage is to discriminate between real and fake samples, researchers recently mentioned two types of classifications namely:

Linear Discriminant Analysis (LDA).

Quadratic Discriminant Analysis (QDA).

Based on our proposed method we extended the classifiers to ensure the quality of our system and in order to report better result using other classifiers, our classifiers where:

Linear Discriminant Analysis (LDA).

Quadratic Discriminant Analysis (QDA).

Logistic Regression (LG).

Linear SVM.

Quadratic SVM.

A brief explanation of the classification metods:

1) Linear Discriminant Analysis (LDA): is defined as the combination of linear features to discriminate between two or more classed by objects or events, this approach is used in machine learning, statistics, and pattern recognition, this method is related to (ANOVA) analysis of variance and regression analysis, (PCA) principal component analysis and factor analysis are similar to linear discriminant analysis because it is used in linear combinations.
2) Quadratic Discriminant Analysis (QDA): is almost similar to linear discriminant analysis, the difference in when using QDA the covariance of each class are not the same, also LDA process for each observation an independent variable unlike QDA.

2.4 Methods Based on Image Quality Features

2.4.1 Methods with Less Than 10 Features

1) Method [25] with 8 quality features:

The paper [25] performs liveness detection, and also confirmed that face recognition is an important field in biometrics, and the importance of this trait for individual's identification.

Due to the existence of spoofing attacks by inserting printed photo, mask, etc. of a genuine individual, this technique weakens the face recognition process, were liveness detection overcomes this problem. By using liveness detection before face recognition some specific features of face that are mainly on the action of eye and mouth are added to the system in a process of increasing security. The proposed liveness module symmetry is tested by using photo, video or mask of a genuine individual.

To perform liveness detection there are three approaches:

1) Using face texture liveness detection.
2) Challenge and response technique for liveness detection
3) Combination of two or more liveness detection

Based on these approaches there are three methods which exist on the field of liveness detection:

1) Multispectral method
2) Client identity information method
3) Single image via diffusion speed model

Based on the existing techniques we can clearly define that under unconstrained environments good results are not obtained in the field of face liveness detection. Hence a proposed method [25] of face liveness detection using image quality assessments (IQA) features is presented.

The proposed method has been validated on a database with images under unconstrained environments. To detect liveness of a face image using image quality assessment features. IQA is used to evaluate the errors extracted from an input image.

There are 8 IQA features used:

SNR: signal to noise ratio [12] equation (2.3)

PSNR: peak signal to noise ratio [11] equation (2.2)

SSI: structural similarity index [17] practical implementation available in [18]

MSE: mean squared error [10] equation (2.1)

TED: total edge difference [14] equation (2.13)

AD: average difference [13] equation (2.6)

NAE: normalized absolute error [13] equation (2.7)

MD: maximum difference [13] equation (2.5)

The proposed technique [25] is designed in the following stages:

1) Query image
2) Enhance
3) Feature extraction
4) Classification

Query image: is the face image input for liveness detection.

Enhance: in this stage a Gaussian filter is applied for filtering noise from the face image and resizing it.

Feature Extraction: in this process image quality assessment is used in calculating features, we considered 8 features for extraction : Peak Signal to noise Ratio (PSNR), Mean Square error (MSE), Normalized Absolute Error (NAE), Signal to Noise Ratio (SNR), Total Edge Difference (TED), Maximum Difference (MD), Structural Similarity Index (SSI), Average

Departure (AD).

Classification: (QDA) quadratic discriminant analysis model is used for classifying if the input image is real or fake.

This system has been tested on a database with 70 face images taken under unconstrained environment.

Table 2.1 shows the proposed method compared with other existing methods, as we can clarify that the IQA method gives indicates:

False Accept Rate (FAR) which indicates the number of false samples classified as real:

FAR= number of fake samples incorrectly accepted as real / total number of images both fake and real. (2.21)

Equation (2.21) is provided in [43]

False Fake Rate (FFR) gives the probability of an image coming from a genuine sample and considered as fake:

FFR= the number of genuine images incorrectly rejected as fake / total number of images both fake and real. (2.22)

Equation (2.22) is provided in [43]

And Half Total Error Rate (HTER) is computed as:

HTER= (FAR+FFR)/2 (2.23)

Equation (2.23) is provided in [43]

These measurements give lower values when compared with other methods based on face liveness detection.

Table 1: Experimental Results Obtained from Different Recognition Methods [25]

Methods	FAR	FFR	HTER
Multispectral	14.98	7.23	18.34
Client identity information	11.96	14.78	21.98
Single image via diffusion speed model	9.23	6.27	11.23
IQA method	6.23	2.19	4.78

2) Method [26] with 6 quality features:

Recent approach [25] is using different identification systems, and machines that satisfies the user’s needs and secure important resource, method [26] reviews biometric identification systems recently developed. This technique is implemented to ensure the identification of an individual weather its real or fake, the aim of this paper is to increase the safety of the biometric system by adding liveness assessment in a user-friendly, fast, simple and non-intrusive manner. This method [26] introduce previous attacks on face, fingerprint, and iris. The proposed method is suitable for real-time applications as it presents a low degree of complexity. This system uses image quality assessments measures extracted from one image to discriminate between real and fake samples. It shows extremely competitive results compared to other existing approaches, when we analyze image quality measures there are valuable information’s that can highly discriminate real samples from impostor traits.

The system [26] objective is to:

1) Evaluate the methodology of protection in multi-biometric dimension, to achieve a better fake detection rate when compared to existing approaches, with different modalities e.g. face, fingerprint, and iris.

2) The ability to notice spoofing attacks and evaluate the methodology of protection in multi-attack dimension.

Based on classification methods used in recent approaches of real and fake samples using LDA and QDA algorithms the present system implements a different approach based on ANN (artificial neural network) algorithm, this algorithm works by loading the entire input query database of images into the program and it operates by comparing it with the database and classifying if the input image is real or fake. The input image is firstly given for feature extraction where the basic IQA features will be calculated then the matcher will classify if the input image is of a genuine user or an impostor client.

In the method [26] six image quality measures are used namely:
Mean Squared Error, Signal to Noise Ratio, Structural Content, Maximum Difference, and Average Difference. After this quality features are calculated an ANN classifier is used together with Feed Forward Neural Network Algorithm in MATLAB 2013 to discriminate between real and fake samples. This method is designed for real time applications with fast, and user-friendly, specifications.

3) Method [27] with 8 image quality features:

The paper [27] is a biometric system used for face image classification, this implemented method uses image quality assessment features to indicate if the input image is real or fake, the proposed method shows that real biometric traits usually gives high valuable information's enough to efficiently discriminate between genuine and impostor traits.

The quality assessment features used in this report are:

Mean Squared Error (MSE) [10] equation (2.1)

Mean Average Error (MAE) [10]

Peak Signal to Noise Ratio (PSNR) [11] equation (2.2)

Structural Content (SC) [13] equation (2.4)

Maximum Difference (MD) [13] equation (2.5)

Normalized Absolute Error (NAE) [13] equation (2.7)

Laplacian Mean Squared Error (LMSE) [13] equation (2.9)

Structural Similarity Index (SSIM) [17] (practical implementation in 18)

This proposed method extracts eight image quality features to discriminate between real and fake samples, it is not mentioned the type of classification method used, this paper also proposed for feature work to increase the multi-biometric system field adding more biometric traits for example signature, palmprint, etc.…

2.4.2 Methods Using 25 Image Quality Feature and Less

1) Method [1] using 25 image quality features:

In [1], a software-based method is used for detecting spoofing attacks, they proposed multiple biometric system that detects face, fingerprint and iris. The objective of this paper is to enhance recognition and protection strategies, to develop the biometric security systems by using image quality assessments and adding liveness assessment in order to improve the quality of speed, make it user friendly and non-intrusive.

The proposed in [1] approach is designed in a suitable manner for real-time applications, with a low degree of complexity, using 25 (IQM)s are extracted from each input image (similar processes used for authentication) in order to discriminate between genuine and fake samples.

The results presented in [1] for face recognition show that their approach is highly competitive compared with other methods and that the use of image quality features extracted from real face samples is very efficient to discriminate them from fake images.

The experimental setup in [1] using Replay-Attack database [40]:
Using a 64-bit windows 7 pc with MATLAB 2012b and Replay-attack database [40] contains 50 different subjects collected from 10 second videos acquired using 320 * 240 resolution webcam of a MacBook Laptop. Results were tested based on a printed spoof attack under specific conditions like a hand holding the picture, fixed picture and both. Researchers also took into consideration the execution time. This results were reported in term of standard rates Table 2.1. FFR is defined as the probability of incorrectly considering a genuine sample as fake equation is (2.22), FGR gives the number of fake images that are classified as real equation is (2.21) (FGR = FFR), and HTER is computed as the average of both FFR and FGR; HTER=(FFR+FGR)/2.

The results reported under hand based condition where FFR=13.6, FGR=5, HTER=9.3, and the results reported on a fixed based condition where FFR=11.5, FGR=5.3, HTER=8.4. And an experiment on a mixture of hand and fixed conditions together gave results as FFR=11.6, FGR=4.1, HTER=7.9. With an average execution time of 0.148 seconds for all conditions, the present research also reported some results based on the best IQA features used with best-5, best-10, and best-15 compared with all the 25 IQA metrics.

Table 2: Experiments Done on Different Number of Features [1]

	Measures	HTER
Best-5	NCC [13],RAMD [10],MAS [10],SPE [15],RRED [20]	53.5
Best-10	MSE [10],AD [13],SC [13],NCC [13],MD [13], RAMD[10], MAS[10], SME[15], SPE[15]	48.9
Best-15	MSE [10],PSNR [11],AD [13],SC [13],NCC [13],MD [13], SNR[12],RAMD[10],MAMS[10],SME[15],SPE[15], TCD[14],GME[16],VIF[19],NIQE[24]	38.3
All	ALL	15.2

From Table 2 [1], we see that there wasn't any clear method of choosing best features; some features are present in best-5 and not in best-10, which shall be investigated in our proposed method. In addition to this, this approach [1] was also compared to some existing methods based on printed spoofing attacks.

Table 3: Comparison between Method and Other State-of-art Methods Based of Spoofed Printed Face Detection [1]

METHODS	FFR	FGR	HTER
IQA-based [1]	0.0	1.0	0.5
AMILAB [32]	0.0	1.2	0.6
CASIA [32]	0.0	0.0	0.0
IDIAP [32]	0.0	0.0	0.0
SIANI [32]	0.0	21.2	10.6

UNICAMP [32]	1.2	0.0	0.6
UOULU [32]	0.0	0.0	0.0

Based on [1] the reported results (Table 3) it is clear that the IQA-based method did not give 100% positive identification at the other hand CASIA, IDIAP, and UOULU methods gave a perfect identification rate with 0% of (FFR) and (FGR).

2) Methods using 18 image quality features:

The paper [28] introduces REPLAY-MOBILE database [41], and compares existing face recognition approaches based on (IQA) image quality assessment measures, this method also provides a number of classifiers to discriminate between real and impostor samples. Based on the existing method 2-sets [1], [33] of presentation attack detection (PDA) results are presented on face recognition based on image quality assessment, the results are presented on ISO standard metrics [see the ISO/IEC 30107-3 standard], (APCER) Attack Presentation Classification Error Rate; and (BPCER) Bona fide Presentation Classification Error Rate.

This proposed paper compares 2-sets of presentation attack detection (PDA) results based on face recognition and classification, Face-PAD using IQA [1], and Face-PAD based on Gabor-Jets [33].

Face-PAD using IQA: the experiments conducted on this paper are based on 18 image quality measures and tested using Replay-Mobile database [41]

The quality features calculated are:

Mean Squared Error MSE [10] equation (2.1)

Peak Signal to Noise Ratio PSNR [11] equation (2.2)

Average difference AD [13] equation (2.6)

Structural content SC [13] equation (2.4)

Normalized cross-correlation NK [13] equation (2.10)

Max. Difference MD [13] equation (2.5)

Laplacian MSE LMSE [13] equation (2.9)

Normalized Absolute error NAE [13] equation (2.7)

Signal to noise ratio SNR [12] equation (2.3)

R-averaged Max. Difference (r=10) RAMD [10] equation (2.8)

Mean angle similarity MAS [10] equation (2.11)

Mean angle magnitude similarity MAMS [10] equation (2.12)

Spectral magnitude error SME [15] equation (2.15)

Gradient magnitude error GME [16] equation (2.18)

Gradient phase error GPE [16] equation (2.19)

Structural similarity index SSIM [17] practical implementation [18]

Visual information fidelity VIF [19] practical implementation [18]

High-low frequency index HLFI [22] practical implementation [18]

Face-PAD based on Gabor-Jets[33] in this method for feature extraction an approach based on Gabor-Jets has been introduced, the Gabor-Jets has been computed using 40 Gabor wavelets using default parameterization, a process of resizing is introduced to standardize all images to 85×100 pixels, and a retain layer model is presented in processing.

Based on a video database the computed Gabor-jets features are calculated on the face region once the face is detected using a face detector.

Based on the experiments done on these two approaches [2], [33]. The standard ISO rates computed are: (APCER) Attack Presentation Classification Error Rate; and (BPCER) Bona fide Presentation Classification Error Rate.

APCER is considered as False Accept Rate (FAR) and BPCER is False Reject Rate, (ACER) Average Classification Error Rate is also considered as ACER=(APCER+BPCER)/2.

The main difference between these ISO standard rates and the old rates (FAR, FFR, HTER) is that they take into account attacks type, potential and success probability. The PAD algorithm performance can be measured as the lower value of ACER estimates better system performance. Half Total Error Rate (HTER) is also calculated in the presented results.

The method [1] on IQA for face recognition has used Linear Discriminant Analysis (LDA) as a classifier and achieved a result of HTER=15%, , the proposed method [28] used support vector machine (SVM) with radial-basis function(RBF) kernel which presents better face-PAD classification rate than LDA using the same quality measurement features.

The results below in Table 4 present HTER, and EER equal error rate percentage using 2 classification methods Linear discriminant analysis LDA, and Support vector machine radial bias function SVM-RBF , on REPLAY-MOBILE [41] database.

Table 4: Results Presented in Based on Two Different Classifiers [28]

Classification	method [13]	LDA	SVM-RBF

Rate			
Dev.EER (%)		5.06	2.68
Test.HTER (%)	15.20	9.78	5.28

The comparison done in Table 4 is based on PAD protocol and for SVM, LIBSVM implementation has been used with kernel =1.5 (kernel = 1 / # features). The HTER and EER are computed per frame.

Gabor-Jet feature vector using SVM-RBF with kernel = 0.00025, the comparison on the below Table 5 is done based on Replay-Mobile Database.

Table 5: Comparison between Gabor-Jet and IQM Using Different Spoofing Attacks

Classification Rate	HTER (%)	HTER (%)	HTER (%)	HTER (%)	HTER (%)	ACER (%)	APCER (%)	BPCER (%)
Scenarios	MP	MV	PF	PH	GT			
IQM	7.70	13.64	4.22	5.43	7.80	13.64	19.87	7.40
Gabor	8.64	9.53	9.40	8.99	9.13	9.53	7.91	11.15

The scenarios considered in this result Table 5:

MP: matte screen-photo

MV: matte screen-video

PF: print-fixed

PH: print-hand

GT: Grand test

From the results obtained we can come out with the idea that the method based on Gabor-Jet gives better result than that of image quality assessment as both methods were experimented on Replay-Mobile database [41].

3) Methods using 25 image quality features:

In paper [29], they have proposed a biometric system based on iris and face fake detection, several existing methods on liveness detection were adapted and implemented to a limited-constrained scenario. The proposed method is a combination of the feature selection in the existing methods classifiers to perform a classification based on the best features (SVM) support vector machine which is used for training face and iris images.

The input images result as real and fake images by matching with training real and fake samples.

We can describe the present system in the following stages:

1) Input image: the input query image is captured using a sensor, the face should be 2D for image quality assessment calculations.
2) Wiener Filtering [29]: is a filter method used to reduce noise on the input images, the input image I is of size (N×M) will be filtered using a wiener filter and generate a smoothed version of the input image ^I. this filter is adaptive in nature and good for IQA technique.
3) IQ Measures: this measures are divided into (FR) full reference and (NR) no reference, (FR) image quality features depend on the real image that is not distorted, to determine the samples quality.

   In (NR) IQA some pre-trained statistical models are used in estimating the quality of the input image, this depends on a recent knowledge and on the image used for training, features are calculated using the difference in quality between both original image I and smoothed version ^I to estimate the value of (FR) IQA metric. This technique assumes that the quality difference produced using Wiener filter can easily differ between genuine and impostor biometric samples.

2.3.4 SVM Classification

Support vector machines (SVM) are supervised learning models associated learning algorithms used for analyzing data and classifying the input patterns.

SVM Classification Algorithm:

1) Read the input iris or face training images from database.
2) Calculate the 25 image quality assessments full reference and no reference features for the input training images.
3) Combine the 25 quality measures as quality assessment features.
4) Create SVM Classification Training Target and compare the trained features using SVM Classifier.
5) Classify SVM training to two classes and give results of either real or fake image.

2.3.5 Methods Using More Than 25 Image Quality Features

1) Method [30] with 30 image quality measures:

In paper [30] a software-based biometric system is introduced with a multi-attack method in order to improve the biometric system security.

This proposed method is based on image quality assessment to discriminate between real and fake traits. This system presented 30 image quality measurements extracted from the input query image for identifying the user's access attempt; these parameter vectors extracted from the image are classified using linear and quadratic discriminant analysis.

This system adds a liveness assessment technique to ensure the biometric system security and provides a low degree of complexity with good performance. In this multi-biometric system, attacks from face, iris, fingerprints, and hand palm images are detected. In hand palm classification of real or impostor users a discriminating method called Dempster-Shafer theory [35] [34] is used, lots of rotations and translations are presented in hand palm images. Dempster-Shafer method process by combining multiple results of decisions obtained by discriminant analysis and produces decisions between genuine or impostor users.

The aim of [30] is to discriminate between real and fake images. The classifiers used is LDA. The proposed system can be divided into three main parts:

1) The input image is enhanced using a Gaussian filter, and a smoothed version is generated, the quality between the input image and smoothed image is calculated using the image quality assessment metric. This approach considers the loss of quality generated between the original and smoothed image as a quantity to differ between genuine and impostor biometric samples.
2) Feature Extraction: in this part the 30 image qualities measures are extracted and calculated:
3) Classification: this section uses LDA classifier to discriminate between fake and real images.

The results obtained from this work is carried out in terms of False Positive Rate (FPR) which indicates the number of false samples classified as real equation is given (2.21) and True Negative Rate (TNR) that gives the probability of an image coming from a genuine sample and considered as fake equation is given (2.22) .

The results obtained from face where classified using (LDA) Linear Discriminant Analysis, the attack considered in this section is printed face photographs, the database consist of 800 samples of real and fake images.

Table 6: Results Reported from Proposed Method [30] Based on Spoofed Printed Faces [30]

FPR	TNR
4.5	8.7

2) Method [31] with 31 image quality features:

The method [31] is developed to increase the biometric security system by using 31 image quality features and adding a liveness assessment method to the system, spoofing attacks is an important field in biometrics, it has been divided into direct and indirect attacks, in this approach these attacks are detected by using 31 IQA and discriminant classifier to discriminate fake and real images, in [30] discriminant power analysis (DPA) is used in face recognition.

The 31 quality features being used in this method are:

Mean Squared Error (**MSE**) [10]

Root Mean Square Error (**RMSE**) [10]

Peak Mean Square Error (**PMSE**) [10]

Mean Absolute Error (**MAE**) [10]

Peak Signal to Noise Ratio (**PSNR**) [11]

Maximum Difference (**MD**) [13]

Signal to Noise Ratio (**SNR**) [12]

Structural Content (**SC**) [13]

Correlation Quality (**CQ**) [13]

Average Difference (**AD**) [13]

Normalized Absolute Error (**NAE**) [13]

R-Averaged Maximum Difference (**RAMD**) [10]

Laplacian Mean Squared Error (**LMSE**) [13]

Error Root Mean Square Contrast (**ERMSC**) [10]

Normalized cross correlation (**NXC**) [13]

Image Fidelity (**IF**) [19]

Mean angle similarity (**MAS**) [10]

Mean angle magnitude similarity (**MAMS) [10]**

Total Edge Difference (**TED**) [14]

Total Corner Difference (**TCD**) [14]

Spectral Magnitude Error (**SME**) [15]

Spectral Phase Error (**SPE) [15]**

Gradient Magnitude Error (**GME**) [16]

Gradient Phase Error (**GPE**) [16]

Structural Similarity Index Measures (**SSIM) [17]**

Visual Information Fidelity (**VIF**) [19]

Reduced Reference Entropic Difference index (**RRED**) [20]

JPEG Quality Index (**JQI**) [21]

High Low Frequency Index (**HLFI**) [22]

Blind Image Quality Index **(BIQI) [23]**

Natural image quality evaluator **(NIQE) [24]**

The system [31] process on a single image it does not require a sequence of images, it also does not require any steps before the computation of image quality features, there are two main stages for this system identification, and authentication.

a) Identification phase consist of:

1) input of image
2) quality features extracted
3) Classification of image either real or fake and output.

Classification process uses three main classifiers, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Artificial Neural Network (ANN), In the identification process the input image is classified using these three classifiers, if the three give positive result as the input image is real the next phase takes step but if one of the classifiers classifies as fake image authentication process does not start.

b) Authentication phase consist of:

1) Discrete Cosine Transform (DCT): DCT is applied to the input face image then using Discriminant Power Analysis (DPA) technique the features considered as the most important are processed.

2) Support Vector Machine (SVM) is used in this stage to discriminate whether the user access is authorized or not

The results reported from these proposed method where experimented on replay attack database for identification and authentication. These experiments where done based on printed faces for three different classifiers:

The results are reported in terms of:

False Accept Rate (FGR) which indicates the number of false samples classified as real Equation is given in (2.21)

False Fake Rate (FFR) that gives the probability of an image coming from a genuine sample and considered as fake: Equation is given in (2.22)

And Half Total Error Rate (HTER) is computed as the average between FFR and FGR the Equation is given in (2.23)

Table 7: Results Presented From the Present Paper [31], Comparison of 3 Classifiers

Classifier	FFR	FGR	HTER
QDA	10.3	8.2	9.25
ANN	5.2	2.1	3.65
LDA	9.2	6.4	7.8

As we can clarify from the above results Table 2.7 ANN classifier gives the best results.

The results reported from authentication approach are based on webcam spoofing attacks (Table 8).

Table 8: Results Reported on SVM Classifier [31]

	FFR	FGR	HTER
SVM	2.2	1.1	1.65

2.3 NUAA Photograph Imposter Database

NUAA Photograph Imposter Database [39], was collected in three sessions with about 2 weeks interval between two sessions, and the place and illumination conditions and scenarios of each session are different as well. Altogether 11 subjects (numbered from 1 to 11) were invited to attend in this work.

Note that it contains various appearance changes commonly encountered by a face recognition system (e.g., sex, illumination, with/without glasses). All original images in the database are color pictures with the same definition of 640 x 480 pixels.

Illustration of different photo-attacks: (1) move the photo horizontally, vertically, back and front; (2) rotate the photo in depth along the vertical axis; (3) the same as (2) but along the horizontal axis; (4) bend the photo inward and outward along the vertical axis; (5) the same as (4) but along the horizontal axis.

In this thesis we will use 600 genuine samples and 700 imposter samples of 11 different users for our test results. Images are resized to 380 × 580.

Type of spoofing attack in NUAA database [39]:

Photograph samples, were taken using high definition photo for each subject using a usual Canon camera in a way that the face area should take at least 2/3 of the whole area of the photograph, then developed the photos in two ways. The first is to use the traditional method to print them on a photographic paper with the common size of 6.8cmx10.2cm (small) and 8.9cm x 12.7cm (bigger), respectively. In the other way, print each photo on an A4 paper using a usual color HP printer.

2.5 Problem Definition

Based on paper [1], we found some problems that will be investigated in this thesis, these problems are:

1) Implement and test real face image detection system (RFIDS)
2) Conduct experiments on RFIDS as in [1]
3) Increase number of classifiers used compared to [1], by trying other classifiers rather than LDA, QDA like Linear SVM, Quadratic SVM, and Logistic Regression.
4) Investigate how to define best 10 and best 5 features that are used but not clearly defined the way of choosing in [1].
5) Compare RFIDS with other methods based on face spoofing attacks [1], [32].
6) Recent papers used different number of quality measures; we are going to investigate the use of 15 image quality measures namely: MES, PSNR, SNR, SC, MD, AD, NAE, RAMD, NCC, TED, TCD, SME, SPE GME, GPE.
7) Examine our proposed method on different data subjects 4, 5, 7, 8 on NUAA database [39]

2.6 Conclusion

In this chapter, we have made a literature survey. From the analysis of [1],[9],[25],[26], [27],[28],[29],[30],[31], we conclude that existing methods use different number of image quality features, and also present different types of classification methods, the results were tested on different databases and we can also say in the recent years that the result obtained were not 100% positive. We defined the problems of the thesis: implement and investigate experimentally real face image detection system (RFIDS)

Chapter 3

IMPLEMENTATION AND TESTING OF RFIDS

RFIDS has two structures training and classification, in our training structure the input is 60 images 30 real and 30 fake and after Gaussian filtering and feature extraction and classification process the output is a training model which we use in our classification structure.

Classification structures input is a sequence of 4 images, we apply Gaussian filter and extract features the input to the classifier is table of faces and training model, according to these inputs classifiers operate and classify our images to either real or fake.

3.1 Training and Detection Structure of RFIDS

Training structure of RFIDS is shown in Figure 2 (a): Training structure of RFIDS (b): Detection Structure of RFIDS

Figure 2 (a) I indicates real image (annotated face image) and `I indicates enhanced image, the input image in RFIDS training structure is a sequence of 60 annotated face images, each image is filtered using a Gaussian filter with 3*3 kernel, then two images are produced the original and enhanced image using Gaussian filter, using these two images the feature extraction process works by extracting 15 image quality features, then a table of annotated face images is created to combine the 60 users and their respective 15 image quality features, in the next stage 5 classification methods are introduced for training the model.

Figure 2 (b) shows RFIDS detection structure, the input to this structure is the sequence of 4 face images for classification process, these images are filtered using Gaussian filter 3*3 kernel, then 15 image quality features are extracted in the feature extraction stage, next a final parameterization is made for combining the 15 image quality features, the final stage is the classification stage where the classifier determines if the images are real or fake depending on the training model.

In the next section of the chapter 3, we give implementation and testing of RFIDS.

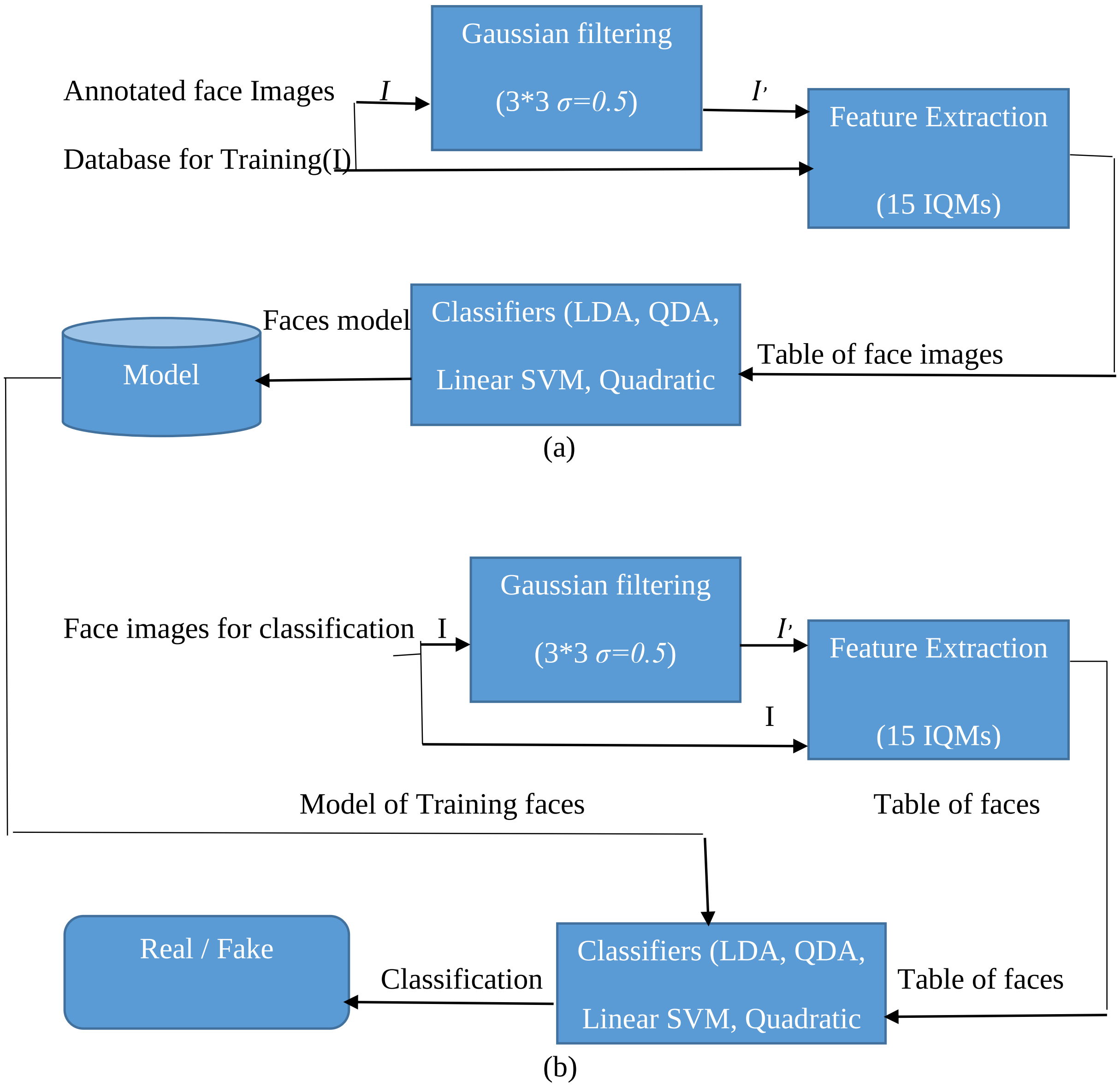


Figure 2: (a) Training Structure and (b) Detection Structure of RFIDS

3.2 Implementation and Testing of Gaussian Filtering

Gaussian filter is use to blur image, it is used to reduce the noise and the image details.

For applying Gaussian blur we have to Design the kernel, the formula to design 2D Gaussian kernel is given by Equations (2.24) and (2.25). A ready matlab function code is available see (Appendix A form line 16-20)

For example:

1) We use a small image to check correctness of Gaussian distribution generation with MATLAB function:
2) The full result screenshots are available in [Appendix C]

Original image:

0	10	7	5
0	2	9	12
4	2	2	6
10	3	9	15

With 0.025 variance and 0 mean:

-0.0018	9.9970	6.9706	4.9659
-0.0171	1.9910	9.0262	12.0202
3.9996	1.9923	1.9888	5.9781
10.0135	3.0052	9.0126	15.0112

With 0.05 variance and 0 mean:

-0.0349	10.0545	7.0376	5.0802

0.0711	1.9968	9.0217	12.0258
4.0066	1.9933	1.9697	5.9810
10.0145	2.9889	8.9668	15.0136

With 0.1 variance and 0 mean:

-0.0379	9.8749	6.9059	5.0493
0.0176	1.9207	9.0281	11.8907
3.8973	1.9792	1.9458	6.0790
9.8374	3.0753	8.9681	15.1888

With 0.5 variance and 0 mean:

0.2382	10.2664	7.0265	4.7215
-0.2709	2.0598	8.8272	12.4882
3.3937	2.5270	2.8774	6.0242
9.0544	2.7450	8.7057	15.7674

With 1 variance and 0 mean:

-2.0454	9.2755	8.2832	4.3307
0.9755	2.8478	8.7432	10.4564
2.9469	1.8501	2.5228	5.2887
8.5677	3.0550	8.2185	15.8684

Table 9: Quality Measures Based on Gaussian Noise

	0	0.025	0.05	0.1	0.5	1
MSE	0	2.9195	0.0013	0.0085	0.2251	1.0158
PSNR	INF	83.477	76.903	68.815	54.607	48.062
SNR	INF	52.740	46.166	38.078	23.870	17.325
SME	0	0.0038	0.0064	0.1043	2.6984	9.2138
SPE	0	2.7666	7.974	3.2452	0.0022	0.0018
GME	0	4.7342	0.0015	0.0139	0.3507	1.0240
GPE	0	1.1700	1.1194	0.0010	0.0243	0.0531
MD	0	0.0341	0.0802	0.1888	0.9456	2.0454
SC	1	0.9993	0.9964	1.0040	0.9846	1.0694
AD	0	0.0029	-0.011	0.0237	-0.028	0.4229
NAE	0	0.0023	0.0048	0.0131	0.0635	0.1450
R-MD	0	0.0232	0..0488	0.1231	0.6473	1.3145
LMSE	0	3.229	5.060	4.893	0.0305	0.0770
NCC	1	1	1.0018	0.9979	1.0058	0.9583

Table 9 shows the results based on Gaussian noise to ensure the correctness of our features calculated we used a small image and to check correctness of Gaussian distribution generation with MATLAB function:

Table 10 : Quality Measures Based on Gaussian Noise

	0	0.025	0.05	0.1	0.5	1
MSE	0	6.256	0.0025	0.0100	0.2503	0.9999
SME	0	73.651	333.905	1.5266	4.8119	2.0300
SPE	0	0.640	0.9221	1.1532	1.4894	1.5722
GME	0	4.853	0.0020	0.0082	0.2303	0.9569
GPE	0	1.917	2.0437	2.1800	2.3926	2.4406
SNR	INF	29.250	23.2174	17.193	3.2297	-2.7856
PSNR	INF	80.167	74.1343	68.110	54.468	48.1315
NCC	1	1.0001	0.9998	1.0001	1.0001	0.9987
AD	0	-7.795	1.2365	-1.5369	-4.416	1.3452
SC	1	0.9987	0.9956	0.9812	0.6777	0.3452
MD	0	0.1210	0.2208	0.4916	2.4653	4.6419
R-MD	0	0.1055	0.2080	0.4384	2.1906	4.2132
NAE	0	0.0334	0.0668	0.1337	0.6666	1.3332
LMSE	0	0.8166	3.2911	13.1390	328.277	1.3096

Table 10 shows the results based on Gaussian noise to ensure the correctness of our features calculated we used a real face image and to check correctness of Gaussian distribution generation with MATLAB function, face image, distribution using Gaussian noise, results and screen shots are provided (see appendix E) Figure E.1 shows original image and Gaussian noise image, Figure E.2 shows corner and edge detection of image, Figure E.3, E.4, E.5, E.6 shows the results implemented with variance 0, Figure E.7 shows original image and Gaussian noise image, Figure E.8 shows corner and edge detection of image, Figure E.9, E.10, E.11, E.12 shows the results implemented with variance 0.025, Figure E.13 shows original image and Gaussian noise image, Figure E.14 shows corner and edge detection of image, Figure E.15, E.16, E.17, E.18 shows the results implemented with variance 0.05, Figure E.19 shows original image and Gaussian noise image, Figure E.20 shows corner and edge detection of image, Figure E.21, E.22, E.23, E.24 shows the results implemented with variance 0.1, Figure E.25 shows original image and Gaussian noise image, Figure E.26 shows corner and edge detection of image, Figure E.27, E.28, E.29, E.30 shows the results implemented with variance 0.05, Figure E.31 shows original image and Gaussian noise image, Figure E.32 shows corner and edge detection of image, Figure E.33, E.34, E.35, E.36 shows the results implemented with variance 1.

3.3 Implementation and Testing of Feature Extraction Subsystem

For testing of the implementation of the features shown below, we are going to use a 4*4 matrix, I(M,N), with M=N=4 to represent a gray scale image to make computation easier and clearer I is original image, ‘I is distorted image.

Original image (reference clean image):

0	10	7	5
0	2	9	12
4	2	2	6
10	3	9	15

Distorted image (smoothed version of the reference image), I(M,N) is as follows:

2	9	10	5
0	1	6	1
3	6	2	6
11	3	14	14

For implementing our 15 image quality assessment features, we refer to respective formula, calculate it manually, show screenshot of the code developed for it, and show and explain the code, the full code is provided in Appendix B.

1) Implementation and testing of Mean Squared Error (MSE): MSE is given by equation (2.1). It is implemented by the following MATLAB code (MSE code see in Appendix B1).

Explanation of code in MSE implementation each numbered line corresponds to its code in Appendix B1:

Line 1 shows the function of mean squared error that we have two inputs realImg corresponds to real image and ehnImg corresponds to enhanced image, Line 4 M, N correspond to the image row and column size respectively of our real image, Line 5 calculates the difference between real and enhanced image, Line 6 calculates the MSE using equation (2.1).

Then for I and ‘I.

MSE= 1/16 * (0-2)^2 + (10-9)^2 + (7-10)^2 + (5-5)^2 + (0-0)^2 + (2-1)^2 + (9-6)^2 + (12-1)^2 + (4-3)^2 + (2-6)^2 + (2-2)^2 + (6-6)^2 + (10-11)^2 + (3-3)^2 + (9-14)^2 + (15-14)^2 =

=1/16*(189) = 11.8125 (3.1)

Results of mean squared error calculation by Code 1 is shown in Figure 3, it complies with (3.1)

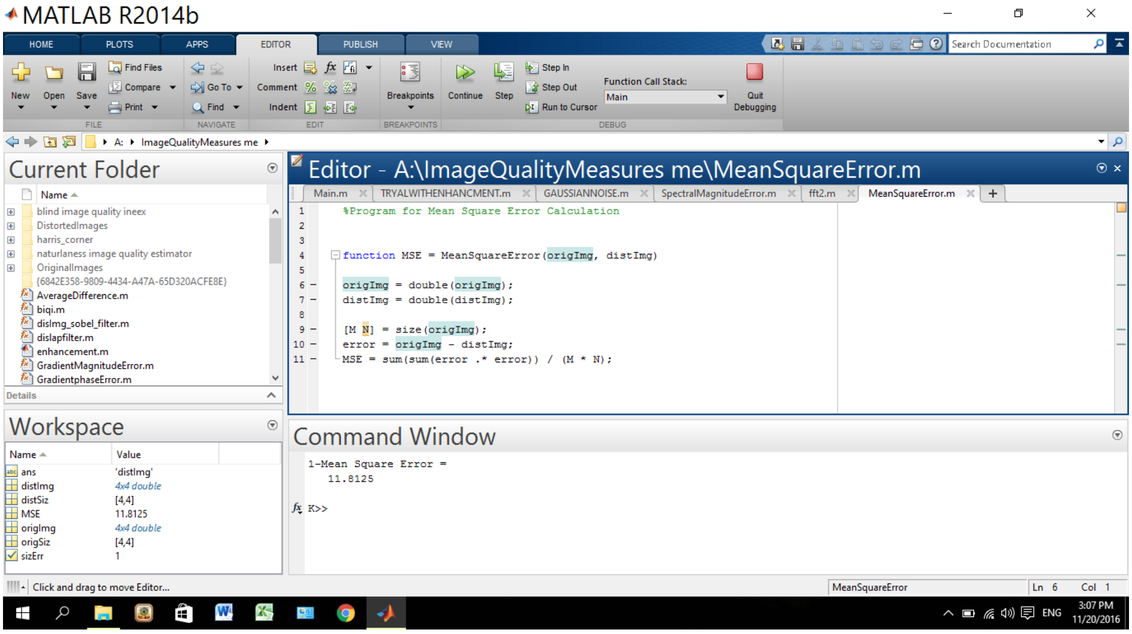


Figure 3: Result Obtained by Code 1 for MSE

2) Implementation and testing of Peak Signal To Noise Ratio (PSNR): PSNR is given by equation (2.2). It is implemented by the following MATLAB code (PSNR code see in Appendix B2).

Code explanation of PSNR implementation each numbered line corresponds to its code in Appendix B2:

Line 1 shows the function of PSNR that we have two inputs realImg corresponds to real image and ehnImg corresponds to enhanced image, Line 4 M, N correspond to the image row and column size respectively of our real image, Line 5 calculates the difference between real and enhanced image, Line 6 calculates the MSE using equation (2.1). Line 7 calculates the PSNR using equation (2.2).

We use MSE=11.8125

MAX I=255 (maximum possible pixel intensity in a grayscale image)

PSNR = 20log MAXI – 10 log MSE

= 20log255 – 10log11.8125

= 48.13 – 10.7

= 37.407 (3.2)

Results of peak signal to noise ratio calculation by Code 2 is shown in Figure 4, it complies with (3.2)

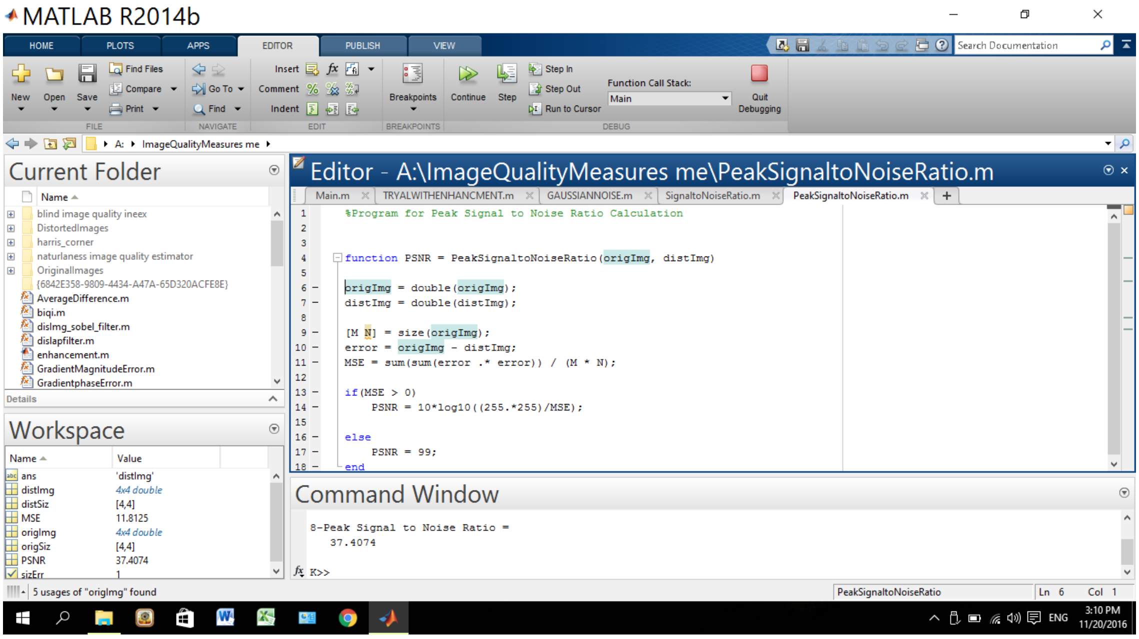


Figure 4: Result Obtained by Code 2 for PSNR (2.2)

3) Implementation and testing of Signal To Noise Ratio (SNR): SNR is given by equation (2.3). It is implemented by the following MATLAB code (SNR code see in Appendix B3).

Code explanation of SNR implementation each numbered line corresponds to its code in Appendix B3:

Line 1 shows the function of SNR that we have two inputs realImg corresponds to real image and ehnImg corresponds to enhanced image, Line 4 M, N correspond to the image row and column size respectively of our real image, Line 5 calculates the difference between real and enhanced image, Line 6 calculates the MSE using equation (2.1). Line 7 calculates the SNR using equation (2.3)

We use MSE=11.8

SNR= 10log (2^2 + 10^2 + 7^2 + 5^2 + 0^2 + 2^2 + 9^2 + 12^2 + 4^2 + 2^2 + 2^2 + 6^2 + 10^2 + 3^2 + 9^2 + 15^2) / 4 * 4 * 11.8

=10 log 882/188.8

=10 log 4.67

=6.6703 (3.3)

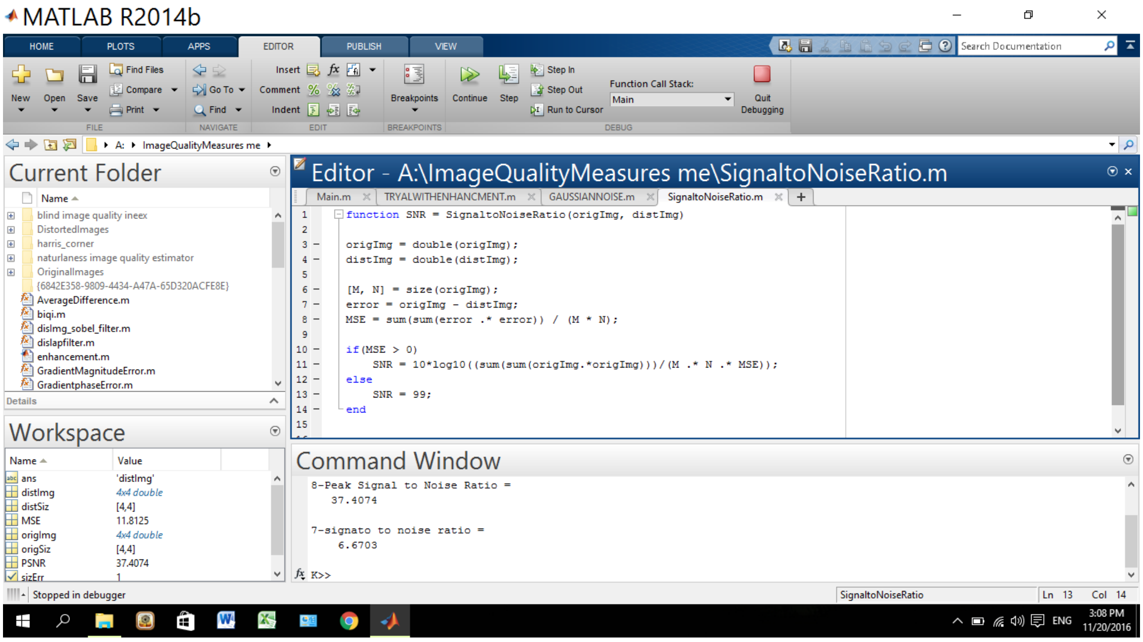


Figure 5: Result Obtained by Code 3 for SNR

4) Implementation and testing of Structural Content (SC): SC is given by equation (2.4). It is implemented by the following MATLAB code (SC code see in Appendix B4).

Code of SC implementation each numbered line corresponds to its code in Appendix B4:

Line 1 shows the function of SC that we have two inputs realImg corresponds to real image and ehnImg corresponds to enhanced image, Line 4 calculates the SC using equation (2.4)

Then for I and ‘I.

SC = (0^2 + 10^2 + 7^2 + 5^2 + 0^2 + 2^2 + 9^2 + 12^2 + 4^2 + 2^2 + 2^2 + 6^2 + 10^2 + 3^2 + 9^2 + 12^2) / (2^2 + 9^2 + 10^2 + 5^2 + 0^2 + 1^2 + 6^2 + 1^2 + 3^2 + 6^2 + 2^2 + 6^2 + 11^2 + 3^2 + 14^2 + 14^2)

=878/855

=1.026 (3.4)

Results of structural content calculation by Code 4 are shown in figure 6. It complies with (3.4)

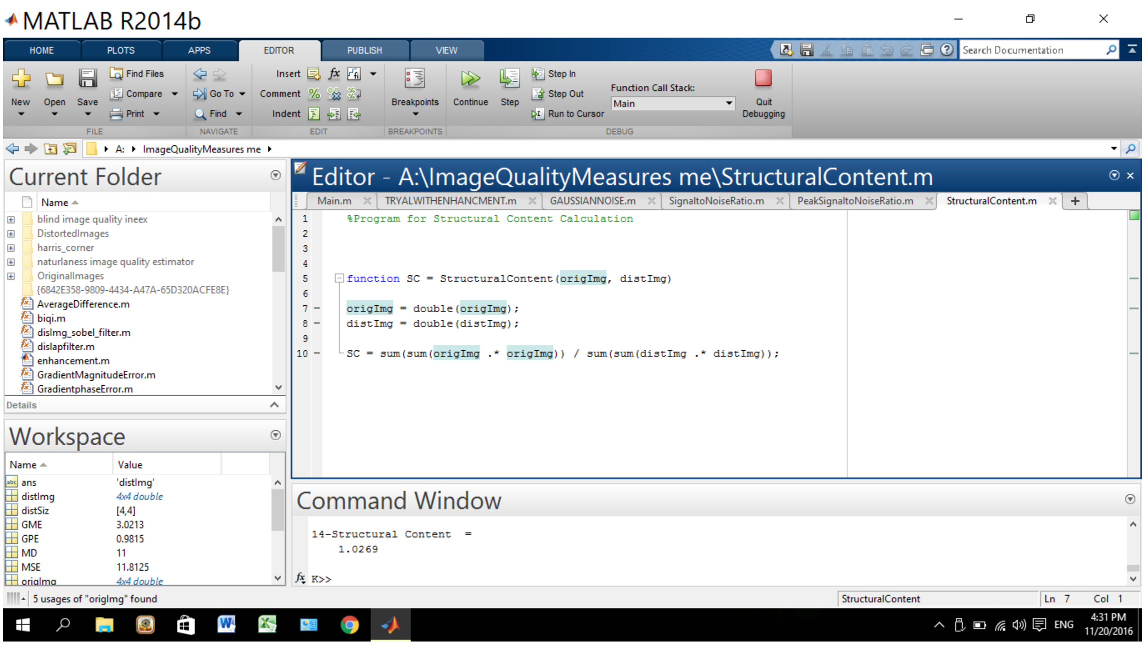


Figure 6: Result Obtained by Code 4 for SC

5) Implementation and testing of Maximum Difference (MD): MD is given by equation (2.5). It is implemented by the following MATLAB code (MD code see in Appendix B5).

Code of MD implementation each numbered line corresponds to its code in Appendix B5:

Line 1 shows the function of MD that we have two inputs realImg corresponds to real image and ehnImg corresponds to enhanced image, Line 4 calculates the difference between real and enhanced image, Line 5 calculates the MD using equation (2.1).

I = reference clean image

$\hat{\imath}$ = smoothed version of the reference image

P1= 0 – 2 = -2	p2= 4 – 3 = 1
P3= 10 – 9 = 1	p4= 2 – 6 = -4
P5= 7 – 10 = -3	p6= 2 – 2 = 0
P7= 5 – 5 = 0	p8= 6 – 6 = 0
P9= 0 – 0 = 0	p10= 10 – 11 = -1
P11= 2 – 1 = 1	p12= 3 – 3 = 0
P13= 9 – 6 = 3	p14= 9 – 14 = -5
p15= 12 – 1 = 11	p16= 15 – 14 = 1

MD=11 (3.5)

Result of Maximum Difference calculation by Code 5 is shown in figure 7. It complies with (3.5)

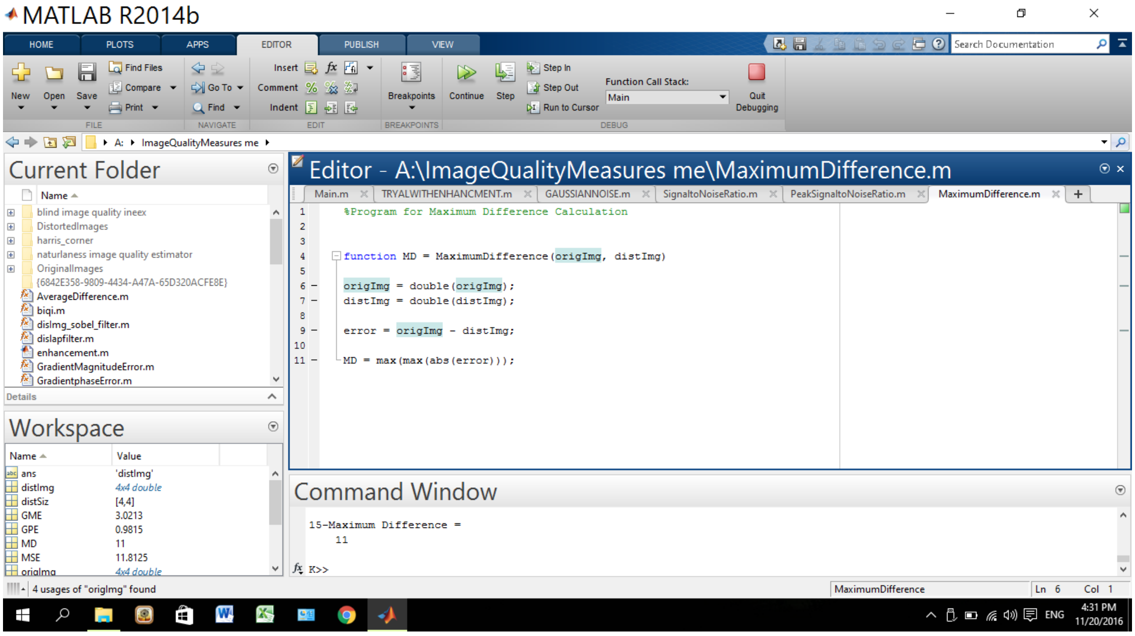


Figure 7: Result Obtained by Code 5 for MD

6) Implementation and testing of Average Difference (AD): AD is given by equation (2.6). It is implemented by the following MATLAB code (AD code see in Appendix B6).

AD code explanation each numbered line corresponds to its code in Appendix B6:

Line 1 shows the function of AD that we have two inputs realImg corresponds to real image and ehnImg corresponds to enhanced image, Line 4 M, N correspond to the image row and column size respectively of our real image, Line 5 calculates the difference between real and enhanced image, Line 6 calculates the AD using equation (2.6)

Then for I, ‘I

AD = 1/16 ((2 - 0) + (9 - 10) + (10 - 7) + (5 - 5) + (0 - 0) + (2 - 1) + (9 - 6) + (12 - 1) + (4 - 3) + (2 - 6) + (2 - 2) + (6 - 6) + (10 - 11) + (3 - 3) + (9 - 14) + (15 - 14))

=1/16(-2 + 1 + -3 + 0 + 0 + 1 + 3 + 11 + -4 + -1 + 0 + -5 + 1 + 1)

=3/16

=0.187 (3.6)

Results of Average Difference calculation by Code 6 is shown in figure 8. It complies with (3.6)

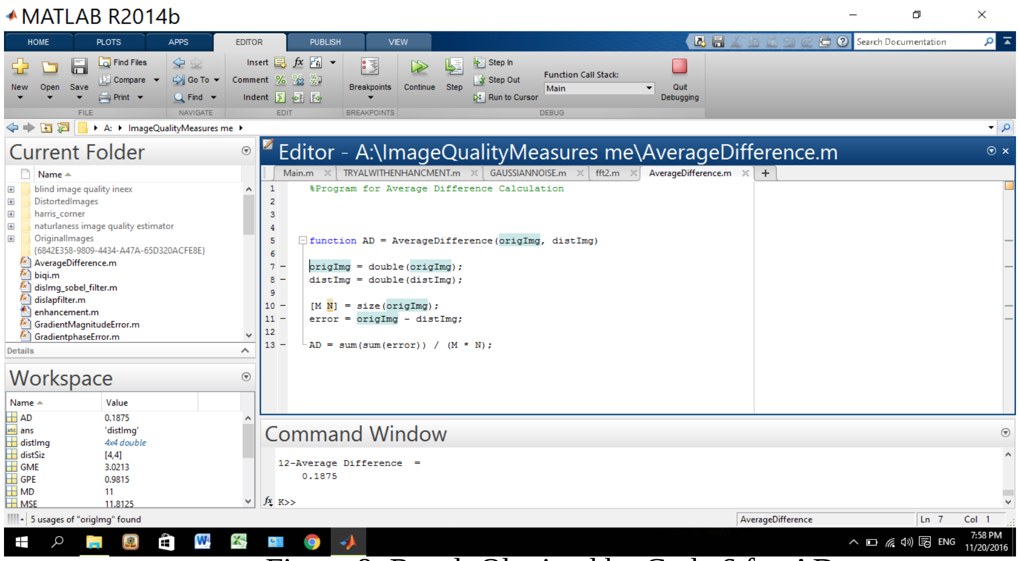


Figure 8: Result Obtained by Code 6 for AD

7) Implementation and testing of Normalized Absolute Error (NAE): NAE is given by equation (2.7). It is implemented by the following MATLAB code (NAE code see in Appendix B7).

Code explanation for NAE implementation each numbered line corresponds to its code in Appendix B7:

Line 1 shows the function of NAE that we have two inputs realImg corresponds to real image and ehnImg corresponds to enhanced image, Line 4 calculates the difference between real and enhanced image, Line 5 calculates the MNAE using equation (2.7).

Then for I,’I.

NAE= | (0 - 2) + (10 - 9) + (7 - 10) + (5 - 5) + (0 - 0) + (2 - 1) + (9 - 6) + (12 - 1) + (4 - 3) + (2 - 6) + (2 - 2) + (6 - 6) + (10 - 11) + (3 - 3) + (9 - 14) + (15 - 14) |

| 0 + 10 + 7 + 5 + 0 + 2 + 9 + 12 + 4 + 2 + 2 + 6 + 10 + 3 + 9 + 15 |

=33 / 96

=0.343 (3.7)

Results of Normalized Absolute Error calculation by Code 7 is shown in figure 9. It complies with (3.7)

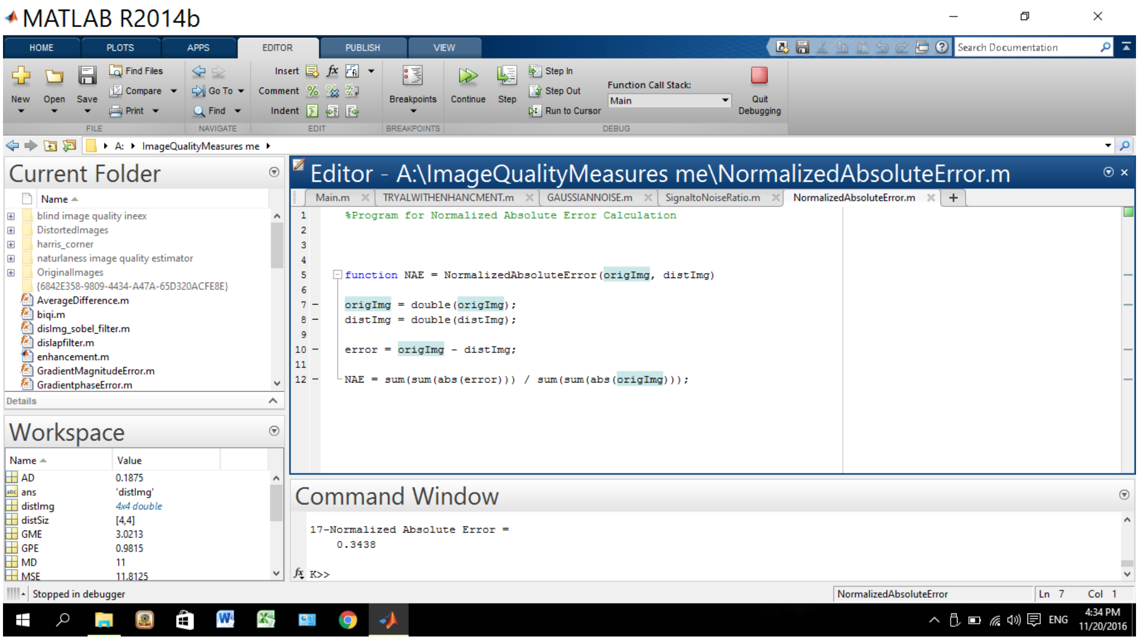


Figure 9: Result Obtained by Code 7 for NAE

8) Implementation and testing of R-Averaged MD: RAMD is given by equation (2.8). It is implemented by the following MATLAB code (RAMD code see in Appendix B8).

Code explanation for RAMD implementation each numbered line corresponds to its code in Appendix B8:

Line 1 shows the function of RAMD that we have two inputs realImg corresponds to real image and ehnImg corresponds to enhanced image

Line 4 Calculate the difference between the real image and enhanced image and save it in error, Line 5 shows how we Assign error as the absolute value of error in Line 6 Convert matrix to vector in Line 7 we Select top ten, because there equals number, there is more than 10 classed. If you want take only 10 include this code

Line 8 shows how we calculate R-averaged MD using equation (2.8).

Then for I,’I.

R = 7 (we take maximum 7 difference between the two images and divide over 7)

P1= 0 – 2 = -2	p2= 4 – 3 = 1
P3= 10 – 9 = 1	p4= 2 – 6 = -4
P5= 7 – 10 = -3	p6= 2 – 2 = 0
P7= 5 – 5 = 0	p8= 6 – 6 = 0
P9= 0 – 0 = 0	p10= 10 – 11 = -1
P11= 2 – 1 = 1	p12= 3 – 3 = 0
P13= 9 – 6 = 3	p14= 9 – 14 = -5
p15= 12 – 1 = 11	p16= 15 – 14 = 1

RAMD =1/7 | 11 + 5 + 4 + 3 + 2 + 1 + 0 |

=26/7

=3.7142 (3.8)

Results of R-Averaged Maximum Difference calculation by Code 8 is shown in figure 10. It complies with (3.8)

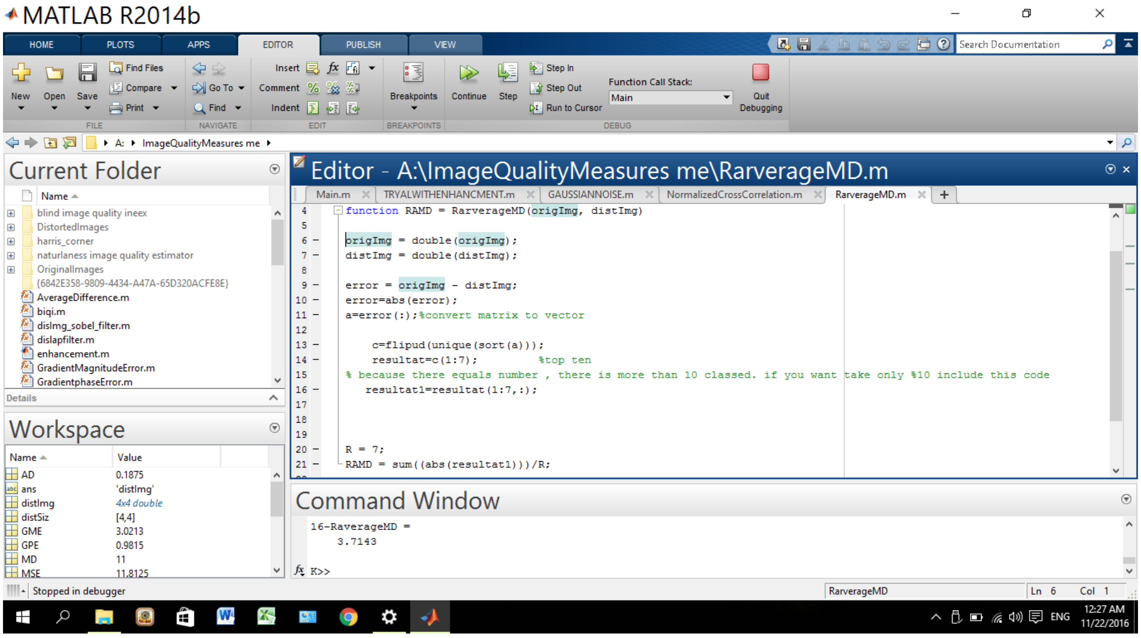


Figure 10: Result Obtained by Code 8 for RAMD

9) Implementation and testing of Normalized Cross Correlation (NCC): NCC is given by equation (2.10). It is implemented by the following MATLAB code (NCC code see in Appendix B9).

Code explanation for NCC implementation each numbered line corresponds to its code in Appendix B9:

Line 1 shows the function of NCC that we have two inputs realImg corresponds to real image and ehnImg corresponds to enhanced image, Line 4 calculates the NCC using equation (2.10).

Then for I,'I.

NXC = (0 * 2) + (10 * 9) + (7 * 10) + (5 * 5) + (0 * 0) + (2 * 1) + (9 * 6) + (12 * 1) + (4 * 3) + (2 * 6) + (2 * 2) + (6 * 6) + (10 * 11) + (3 * 3) + (9 * 14) + (15 * 14) / (0^2 + 10^2 + 7^2 + 5^2 + 0^2 + 2^2 + 9^2 + 12^2 + 4^2 + 2^2 + 2^2 + 6^2 + 10^2 + 3^2 + 9^2 + 15^2)

=(0 + 90 + 70 + 25 + 0 + 2 + 54 + 12 + 12 + 12 + 4 + 36 + 110 + 9 + 126 + 210) / (0 + 100 + 49 + 25 + 0 + 4 + 81 + 144 + 16 + 4 + 4 + 36 + 100 + 9 + 81 + 225)

=760/878

=0.879 (3.9)

Results of R-Averaged Maximum Difference calculation by Code 9 is shown in figure 11. It complies with (3.9).

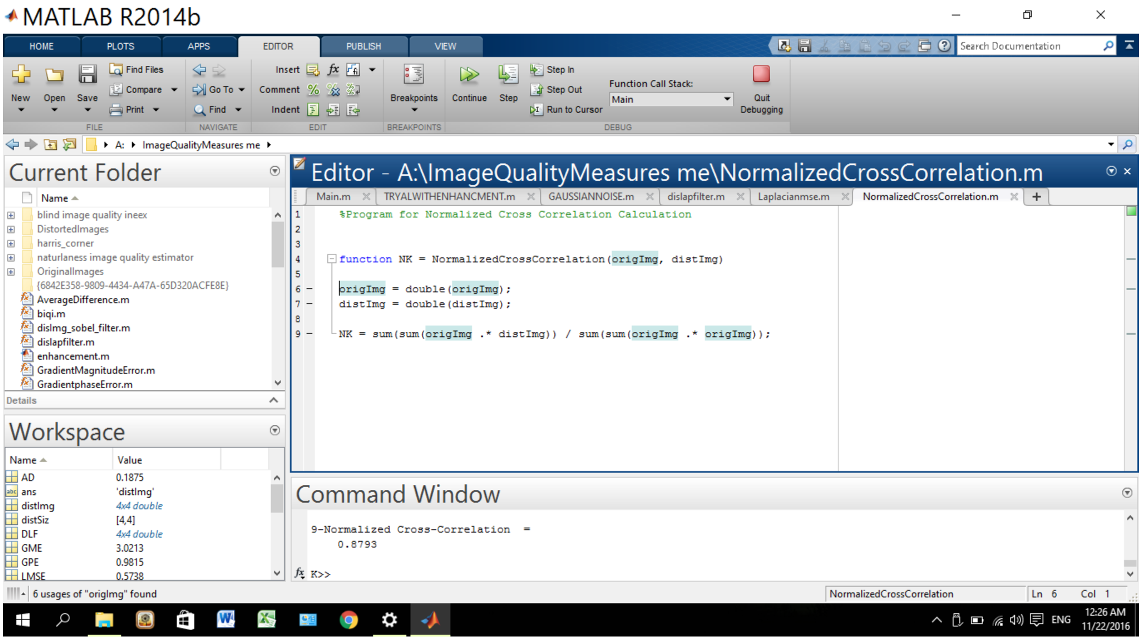


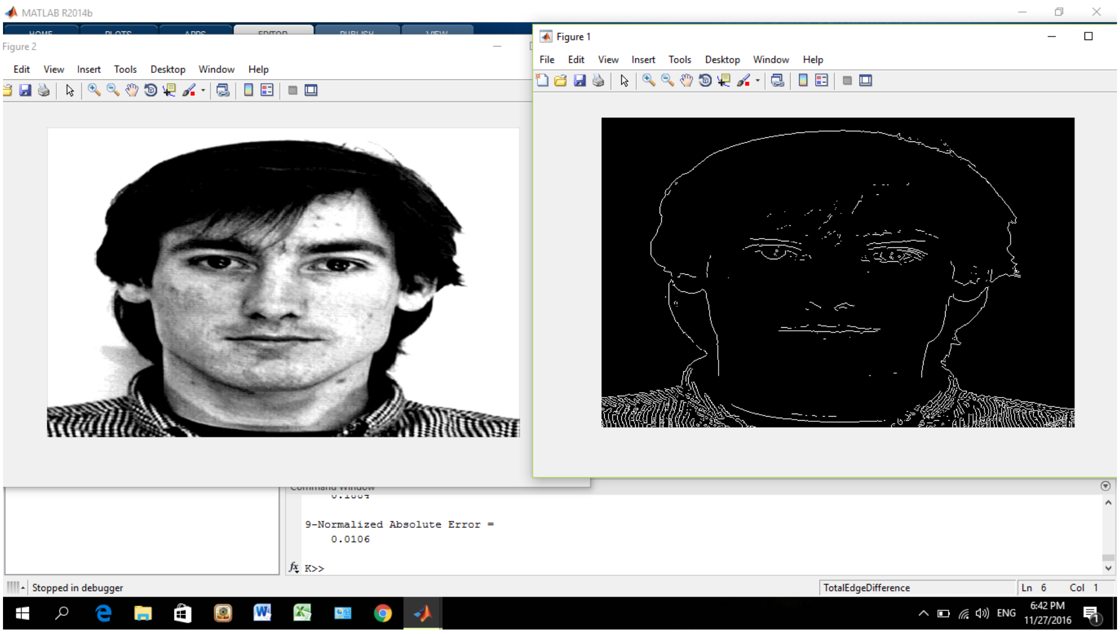
Figure 11: Results Obtained by Code 9 for R-AMD

10) Implementation and testing of Total Edge Difference (14): TED is given by equation (2.13). It is implemented by the following MATLAB code (TED code see in Appendix B10).

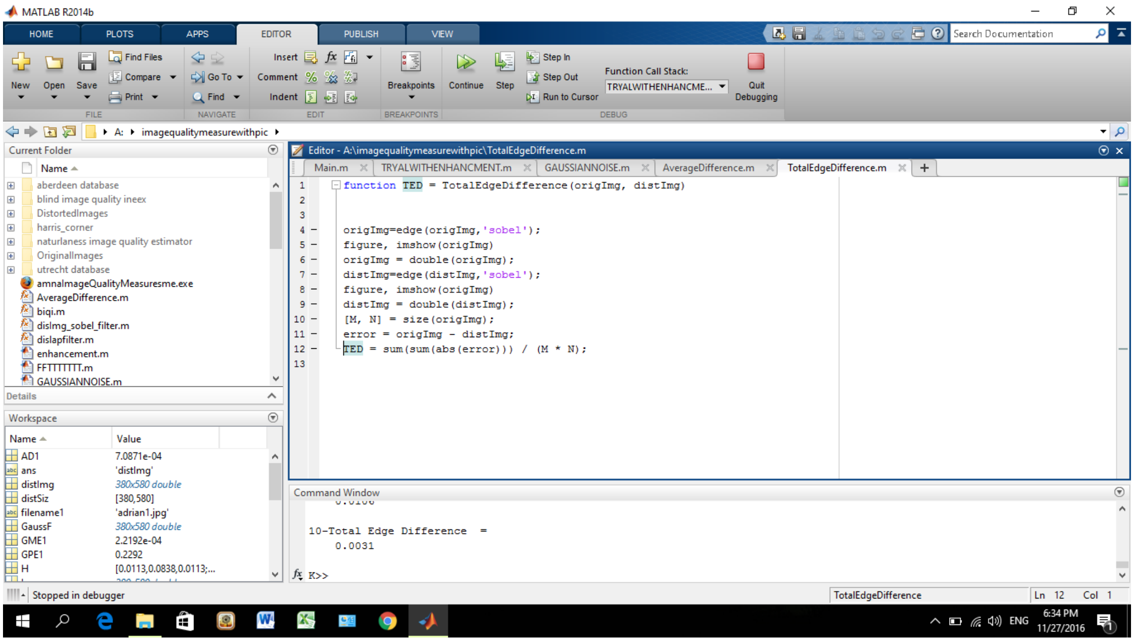
Code explanation for TED implementation each numbered line corresponds to its code in Appendix B10:

Line 1 shows the function of TED that we have two inputs realImg corresponds to real image and ehnImg corresponds to enhanced image, in line 2 we Apply sobel filter to real image in line 5 we Apply sobel filter to enhanced image

Line 8 M, N correspond to the image row and column size respectively of our real image, Line 9 Calculate the difference between the real image and enhanced image and save it in error Line 10 calculates the TED using equation (2.13), Results of Total Edge Difference calculation by Code 10 is shown in figure 12 (a), (b).



(a)



(b)

Figure 12: (a) and (b) Results obtained by Code 10 for TED

11) Implementation and testing of Gradient Magnitude Error: GME is given by equation (2.19). It is implemented by the following MATLAB code (GME code see in Appendix B11).

Code explanation for GME implementation each numbered line corresponds to its code in Appendix B11:

Line 1 shows the function of GME that we have two inputs realImg corresponds to real image and ehnImg corresponds to enhanced image, Line 4 calculates the Gradient transform of image, Line 5 M, N correspond to the image row and column size respectively of our real image, Line 6 Calculate the difference between the real image and enhanced image and save it in error Line 7 calculates the GME using equation (2.19)

Then for I, ‘I.

Original image (reference clean image):

0	10	7	5
0	2	9	12
4	2	2	6
10	3	9	15

Gradient of pixel (2,2)

FX1 = df/dx = 9 – 0 / 2

= 9 / 2 = 4.5

FY1 = df/dx = 2 – 10 /2

= -8 / 2 = -4

Sqrt -4^2 + 4.5^2

= Sqrt 36.25

= 6

FX1:

10	3.5	-2.5	-2
2	4.5	5	3
-2	-1	2	4
-7	-0.5	6	6

FY1:

0	-8	2	7
2	-4	-2.5	0.5
5	0.5	0	1.5
6	1	7	9

Original Image:

10	8.7	3.2	7.2
2.8	6	5.5	3
5.3	1.1	2	4.2
9.2	1.1	9.2	10.2

Distorted Image:

7.2	8.9	4.4	6.4
1.1	3.3	4	5
6.2	1.1	4	7.6
11.3	3.3	13.2	8

Original – Distorted

2.7119	-0.2121	-1.2706	0.8770
1.7104	2.6667	1.5902	-1.9836
-0.8798	0	-2	-3.3602

-2.0942 -2.2361 -3.9808 2.8167

1/16 (2.7199^2 + -0.2121^2 + -1.2706^2 + 0.8770^2 + 1.7104^2 + 26667^2 + 1.5902^2 + -1.9836^2 + -0.8798^2 + 0^2 + -2^2 + -3.3602^2 + -2.0942^2 + -2.2361^2 + -3.9808^2 + 2.8167^2)

=1/16 (7.3978+0.0449+1.6144+ 0.7619 + 2.925 + 7.1112 + 2.5287 + 3.9346 + 0.7740 + 4 + 11.290 + 4.3856 + 5 + 15.8467 + 7.9337)

=4.7223 (3.11)

Results of Gradient Magnitude Error calculation by Code 11 is shown in figure 13. It complies with (3.11)

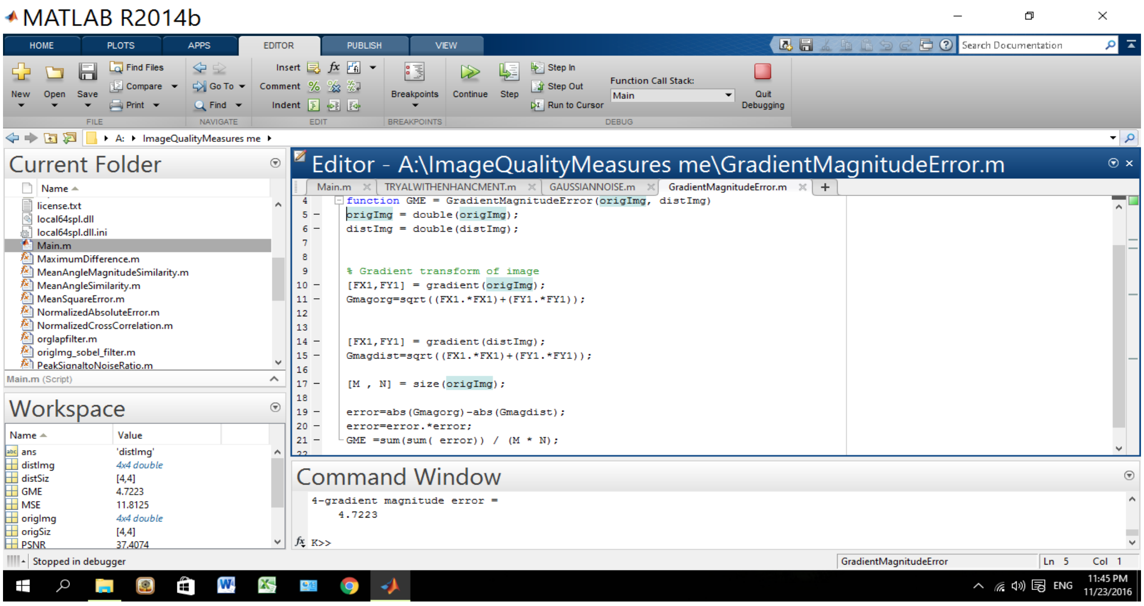


Figure 13: Result Obtained by Code 11 for GME

12) Implementation and testing of Gradient phase error: GPE is given by equation (2.18). It is implemented by the following MATLAB code (GPE code see in Appendix B12).

Code explanation for GPE implementation each numbered line corresponds to its code in Appendix B12:

Line 1 shows the function of GPE that we have two inputs realImg corresponds to real image and ehnImg corresponds to enhanced image, Line 4 show how we calculate Gradient of original image in line5 we Transfer to complex number Line 6 calculate Angle of complex original image Line 7 calculates Gradient of enhanced image, Line 8 Transfer to complex number, Line 9 Angle of complex enhanced image, Line 10 M, N correspond to the image row and column size respectively of our real image, Line 11 Calculate the difference between the real image and enhanced image and save it in error, Line 12 calculates the GPE using equation (2.18).

Then for I, ‘I.

Gradient:

FX1:

10	3.5	-2.5	-2
2	4.5	5	3
-2	-1	2	4
-7	-0.5	6	6

FY1:

0	-8	2	7
2	-4	-2.5	0.5
5	0.5	0	1.5
6	1	7	9

Z=

10 + 0i	3.5 – 8i	-2.5 + 2i	-2 + 7i
2 + 2i	4.5 – 4i	5 – 2.5i	3 + 0.5i
-2 + 5i	-1 + 0.5i	2 + 0i	4 + 1.5i
-7 + 6i	-0.5 + 1i	6 + 7i	6 + 9i

F=

7-2i	4-8i	-2- 4i	-5-4i
1+0.5i	3-1.5i	0-4i	-5+0.5i
3+5.5i	-0.5+1i	0+4i	4+6.5i
-8+8i	1.5-3i	5.5+12i	0+8i

arg original (angle(z)):

0	-1.1584	2.4669	1.8491

arg distorted (angle(f)):

-0.27	-1.107	-2.034	-2.466

0.785	-0.726	-0.463	0.165		0.463	-0.463	-1.570	3.041
1.9513	2.6779	0	0.3588		1.071	2.034	1.570	1.019
2.4330	2.0344	0.8622	0.9828		2.366	-1.107	1.141	1.570

arg(original) – arg(distorted)

-0.2783	0.0512	0.4324	-0.6178
0.3218	0.2630	-1.1071	-2.8768
0.8799	0.6435	-1.5708	-0.6604
0.0768	0.9273	-0.2789	-0.588

GPE = 1/16 (-0.2783^2 + 0.0512^2 + 0.4324^2 + -0.6178^2 + 0.3218^2 + 0.2630^2 + -1.1071^2 + -2.8768^2 + 0.8799^2 + 0.6435^2 + -1.5708^2 + -0.6604^2 + 0.0768^2 + 0.9273^2 + -0.2789^2 + -0.588^2)

GPE = 4.7223 (3.12)

Results of Gradient phase error calculation by Code 12 are shown in Figure 14. It complies with (3.12)

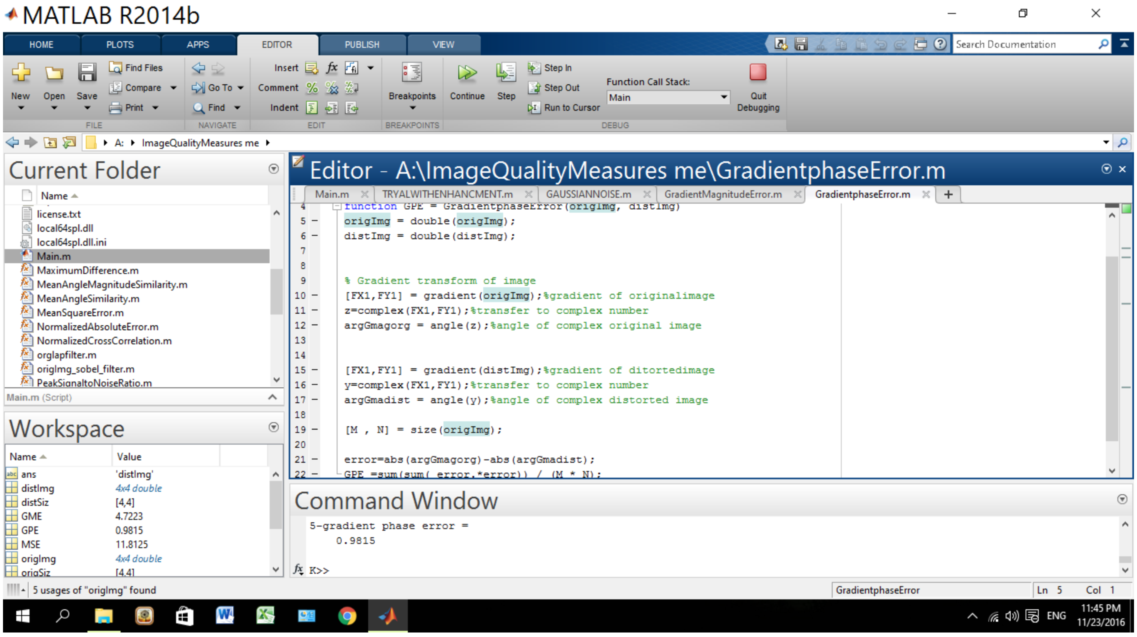


Figure 14: Result Obtained by Code 12 for GPE

Implementation and testing of Spectral Magnitude Error: SME is given by equation (2.16). It is implemented by the following MATLAB code (SME code see in Appendix B13).

Code explanation for SME implementation each numbered line corresponds to its code in Appendix B13:

Line 1 shows the function of SME that we have two inputs realImg corresponds to real image and ehnImg corresponds to enhanced image, Line 4 Fast Fourier transform of image, Line 5 shows the Image real part, Line 6 shows Image imaginary part, Line 7 calculates Gradient of image, Line 8 shows Image real part, Line 9 shows Image imaginary part, Line 10 calculates Gradient of image

Line 11 calculates the SME using equation (2.16).

Then for I, ‘I.

2D-Fourier Transform Equation [39]:

Original image:

0	10	7	5
0	2	9	12
4	2	2	6
10	3	9	15

FFT Original image:

96+0i	-13+21i	-14+0i	-13+21i
8+14i	-11+1i	-6+6i	-7+19i
-2+0i	3-23i	-6+0i	3+23i
8-14i	-7-19i	-6-6i	-11-1i

Distorted image:

2	9	10	5
0	1	6	1
3	6	2	6
11	3	14	14

FFT Distorted image:

93+0i	-16+7i	3+0i	-16-7i
9+34i	-20-1i	5+4i	2+7i
-7+0i	2-15i	-21+0i	2+15i
9-34i	2-7i	5-4i	-20+1i

FFT (Original Image) Gradient:

Sqrt 96^2 + 0^2

= 96

Sqrt -13^2 + 21^2

= 24.698

Gradient (Original Image):

96	24.698	14	24.698
16.124	11.045	8.485	20.248
24	23.194	6	23.194
16.124	20.248	8.485	11.045

Gradient (Distorted Image):

93	17.464	3	17.464
35.171	20.025	6.4031	7.280
7	15.132	21	15.132
35.171	7.280	6.403	20.025

Original – distorted

3	7.233	11	7.232
-19.046	-8.979	2.082	12.968
17	8.062	-15	8.062
-19.046	2.968	2.082	-8.979

SME = 1/16 (3^2 + 7.233^2 + 11^2 + 7.232^2 + -19.046^2 +-8.979^2 + 2.082^2 + 12.968^2 + 17^2 + 8.062^2 + -15^2 + 8.062^2 + -19.046^2 + 2.968^2 + 2.082^2 + -8.979^2)

= 131.905 (3.13)

Results of Spectral Magnitude error calculation by Code 13 is shown in Figure 15. It complies with (3.13)

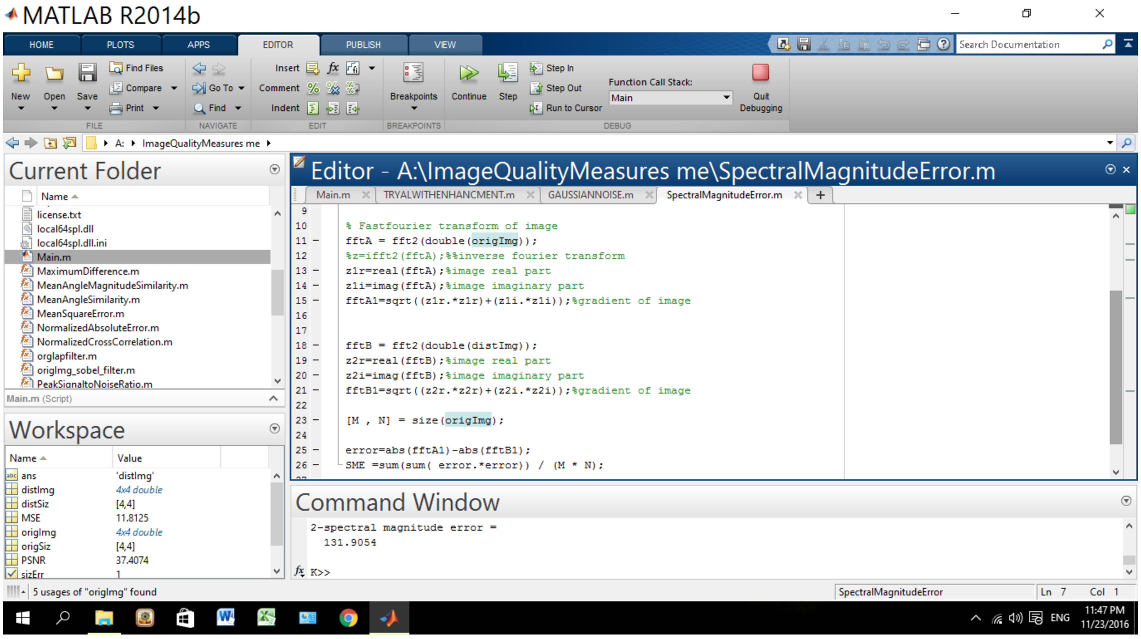


Figure 15: Results Obtained by Code 13 for SME

13) Implementation and testing of Spectral phase error: SPE is given by equation (2.15). It is implemented by the following MATLAB code (SPE code see in Appendix B14).

Code explanation for SPE implementation each numbered line corresponds to its code in Appendix B14:

Line 1 shows the function of SPE that we have two inputs realImg corresponds to real image and ehnImg corresponds to enhanced image, Line 4 calculates Fast Fourier transform of image.

Line 1 shows the function of SPE that we have two inputs realImg corresponds to real image and ehnImg corresponds to enhanced image, Line 4 M, N correspond to the image row and column size respectively of our real image, Line 5 calculates the difference between real and enhanced image, Line 5 calculates the SPE using equation (2.15).

Then for I, ‘I.

Original image:				FFT Original image:			
0	10	7	5	96+0i	-13+21i	-14+0i	-13+21i
0	2	9	12	8+14i	-11+1i	-6+6i	-7+19i
4	2	2	6	-2+0i	3-23i	-6+0i	3+23i
10	3	9	15	8-14i	-7-19i	-6-6i	-11-1i

Distorted image:				FFT Distorted image:			
2	9	10	5	93+0i	-16+7i	3+0i	-16-7i
0	1	6	1	9+34i	-20-1i	5+4i	2+7i
3	6	2	6	-7+0i	2-15i	-21+0i	2+15i
11	3	14	14	9-34i	2-7i	5-4i	-20+1i

Argument (original image)				Argument (distorted image)			
0	2.125	3.141	-2.125	0	2.729	0	-2.729
1.051	3.050	2.356	1.923	1.312	-3.091	0.674	1.292
3.141	-1.441	3.141	1.441	3.141	-1.438	3.141	1.438
-1.051	-1.923	-2.356	-3.050	-1.312	-1.292	-0.674	3.019

Original – distorted

0	-0.604	3.141	-0.604
-0.260	-0.040	1.681	0.631
0	0.002	0	0.002
-0260	0.631	1.681	-0.040

$$SPE = 1/16\ (0^2 + -0.604^2 + 3.141^2 + -0.604^2 + -0.260^2 + -0.040^2 + 1.681^2 + 0.631^2 + 0^2 + 0.002^2 + 0^2 + 0.002^2 + 0260^2 + 0.631^2 + 1.681^2 + -0.040^2)$$

$$SPE = 1.074 \tag{3.14}$$

Results of Spectral phase error calculation by Code 14 are shown in Figure 16. It complies with (3.14)

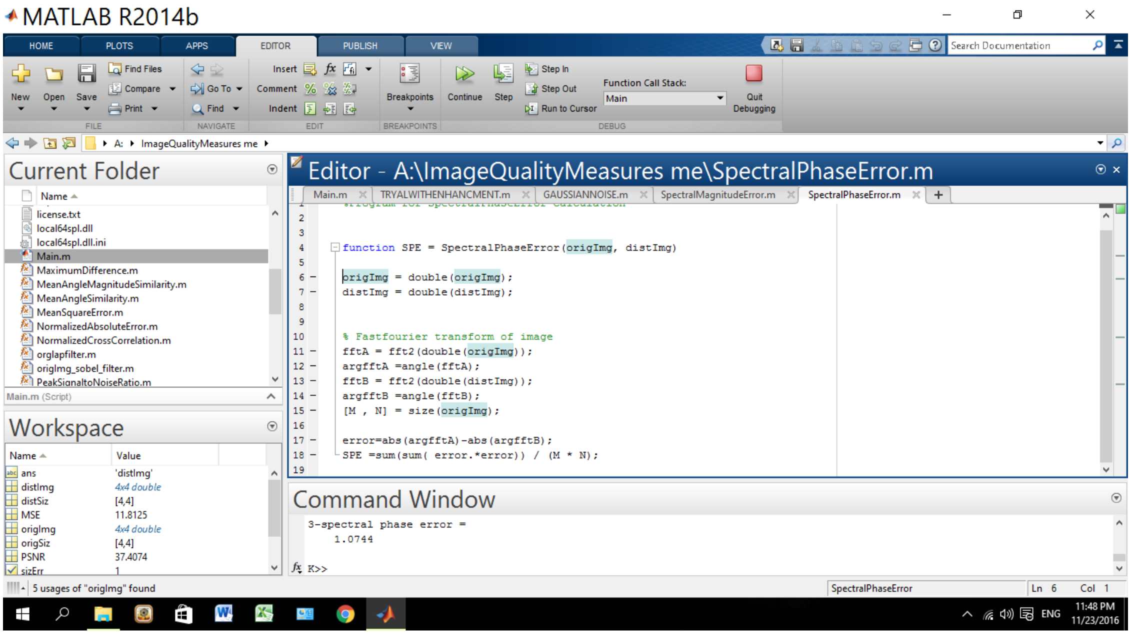


Figure 16: Result Obtained by Code 14 for SPE

14) Implementation and testing of Total corner difference: TCD is given by equation (2.14). It is implemented by the following MATLAB code (TCD code see in Appendix B15).

Code explanation of TCD implementation each numbered line corresponds to its code in Appendix B15:

Line 1 shows the function of TCD that we have two inputs realImg corresponds to real image and ehnImg corresponds to enhanced image, in line 4 we Apply Harris corner detector to real image, line 5 we Plot number of corners In line 6 we Calculate the number of corners detected by Harris detector in line 7 Apply Harris corner detector to enhanced image, line 8 Plot number of corners, in line 9 we Calculate the number of corners detected by Harris detector, line 10 we Select the maximum number of corners between original and enhanced images, Line 11 calculates the TCD using equation (2.14).

N_{cr} = number of corners in original image using Harris corner detector

$\hat{N}_{cr}$ = number of corners in distorted image using Harris corner detector

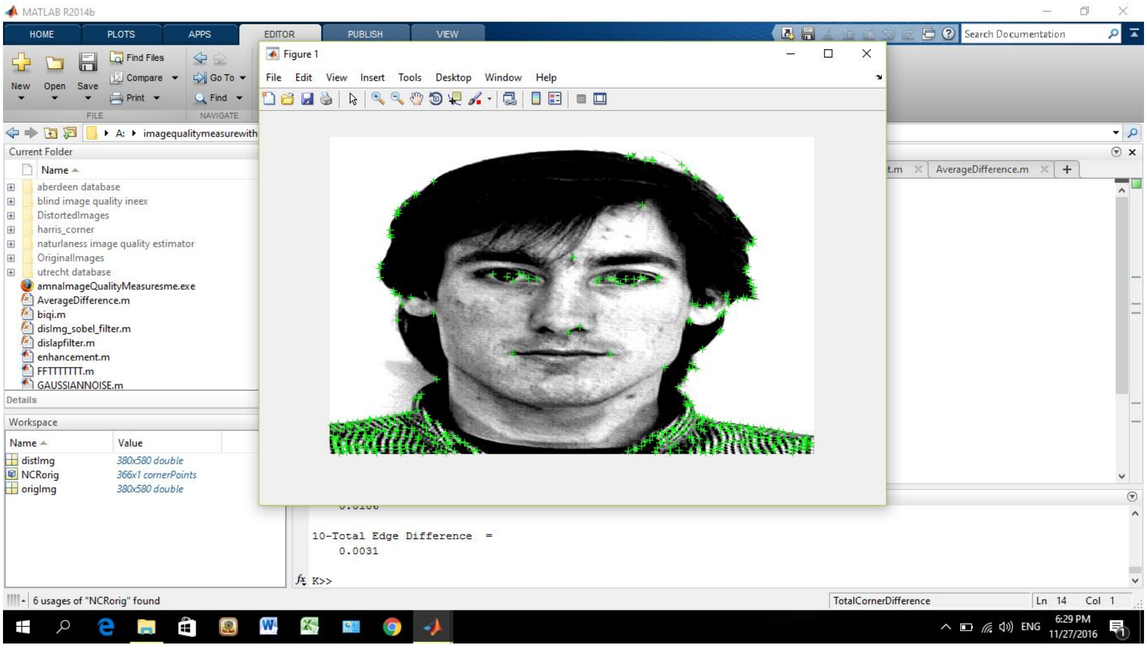
N_{cr} = 366

$\hat{N}_{cr}$ = 385

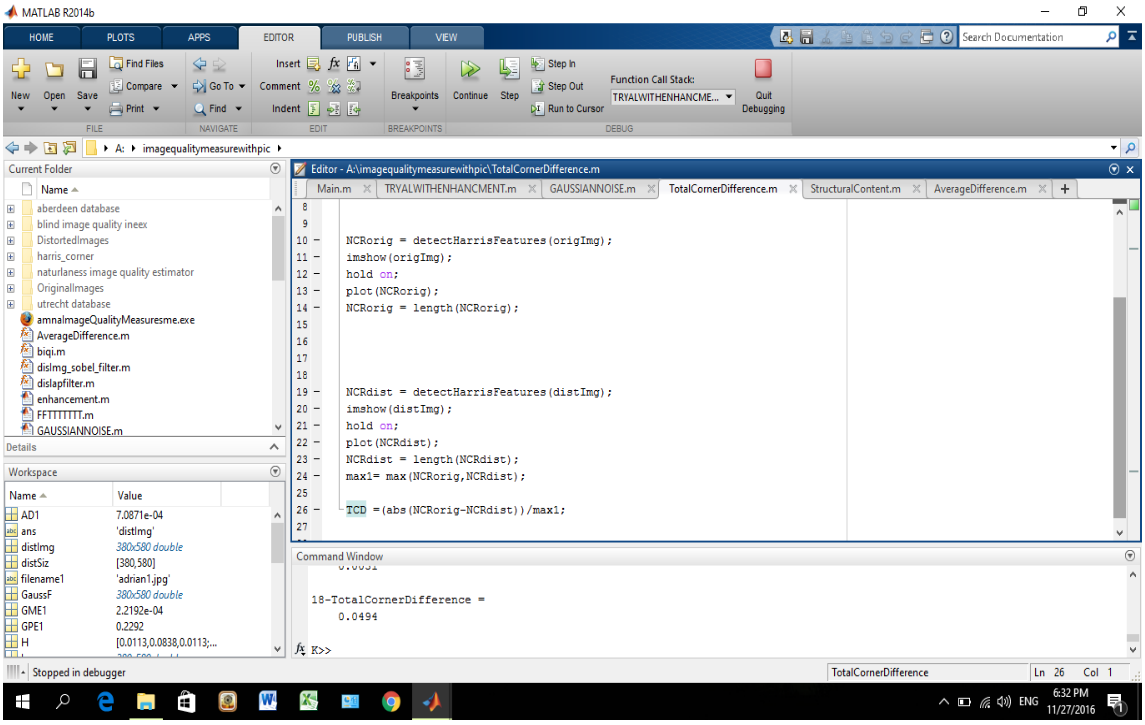
TCD = |366 – 385| / 385

= 0.0494 (3.15)

Results of mean squared error calculation by Code 15 are shown in Figure 17. It complies with (3.15)



(a)



(b)

Figure 17: (a) and (b) Results Obtained for TCD

3.4 Implementation of Training Structure

In MATLAB 2016 there is a ready application provided called classification learner that we can use to import tables from our work space, these application extracts predictors and observations and allows a number of classification algorithms (LDA, QDA, Linear SVM, Quadratic SVM, Logistic Regression) to train the samples extracted.

For training we used NUAA database as we considered 60 face image samples, 30 real and 30 fake images see (Appendix C) for samples feature values, the training process in a model for which the classification process relay on.

Classification learner application in MATLAB: Using the classification learner application for training we have to arrange our features results into a table and display them in workspace, the classification learner app imports all results in work space and asks for permission of which table you want to use, we select the table T containing all results of implemented faces with all 15 features, the next step will be the selection of response and predictors the response in our implementation is real or fake, where we have 15 predictors that are the quality features, then we run our training process to train the samples imported to obtain the model we use in classification process for either classifying face image as real or fake, The following screen shots Figures 18 - 23 show the steps on how the training process work:

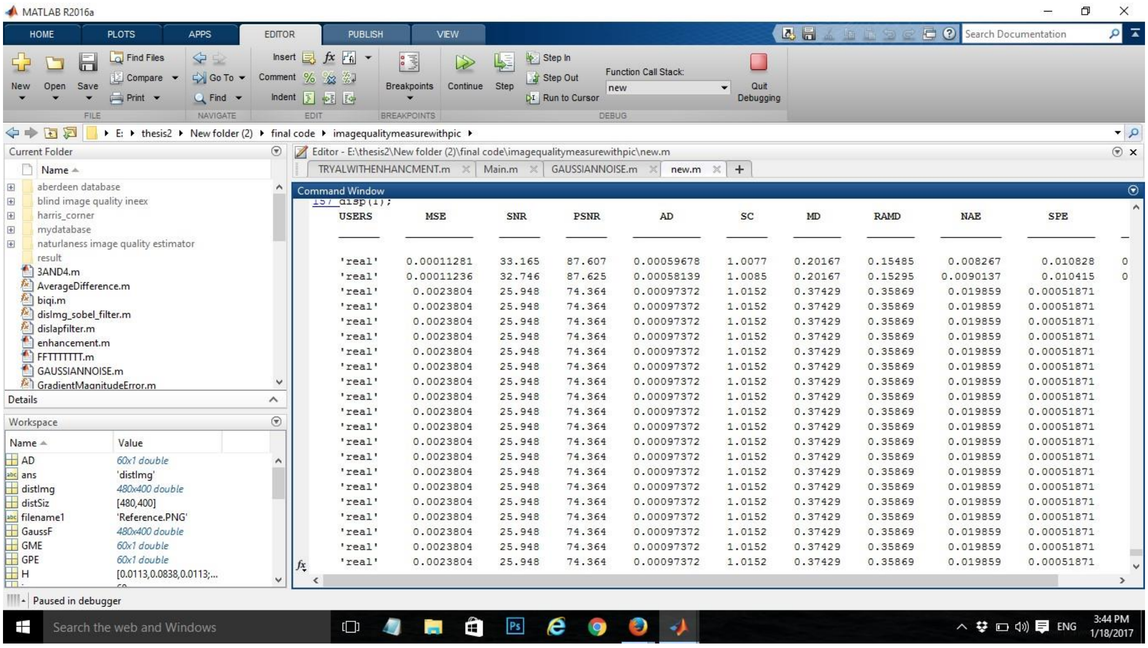


Figure 18: The Results of Faces are Created in the Workspace

Figure 18 the left side shows the matlab files for calculation of our features and the middle side shows the table T arranged for using 15 features and 60 face image for one user, 30 real and 30 fake images, it does not show all 60 arranged users due to limitation of screenshot, there are 15 features shown and real and fake users for 60 users full features calculations see (Appendix C). Table C.1, C.2, C.3 shows the calculations of 15 image quality features of 60 users, 30 real and 30 fake users.

It also shows how the table is created in the workspace for importing this table in our classification learner application, it is not complete due to limitation of screenshot.

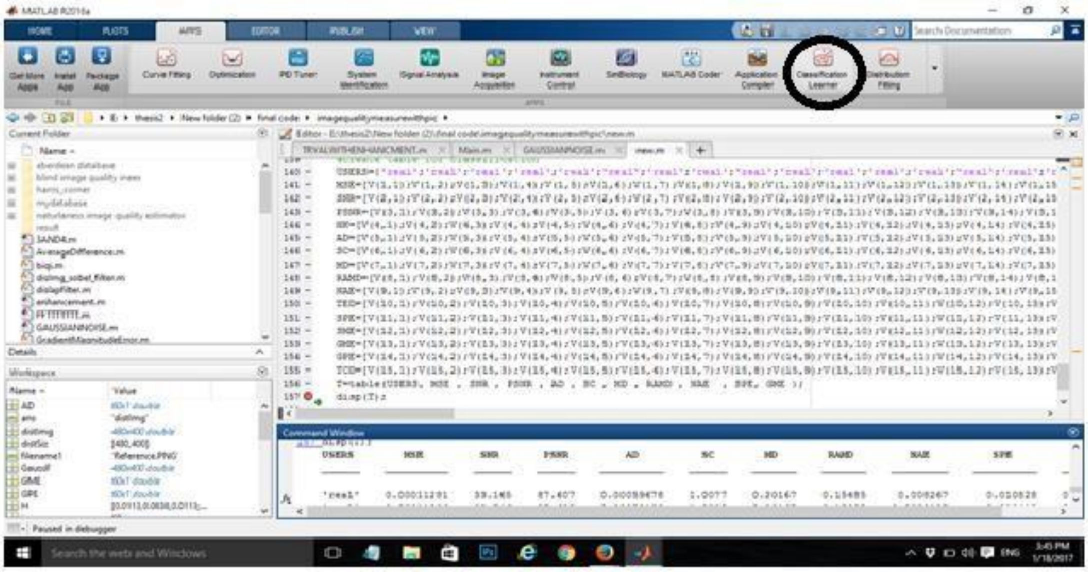


Figure 19: The application Used is Classification Learner App which is Clear above

Figure 19 shows the application of MATLAB we used classification learner, click on APP and select classification learner.

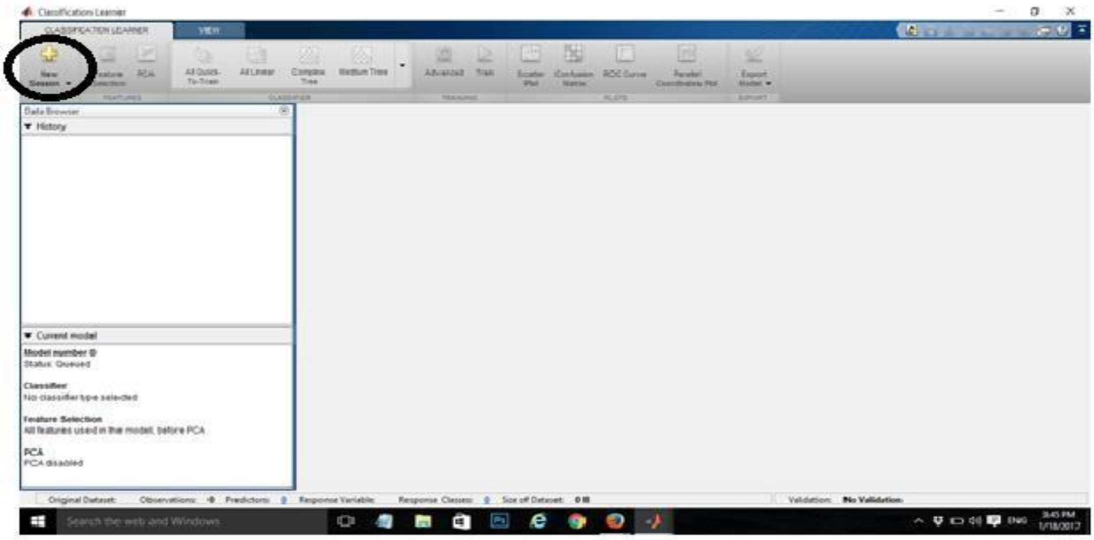


Figure 20: Run of Classification Learner Application and Click New Session to Import Faces from Work Space

Figure 20 shows the first page of classification learner application, on this page the only clickable choice is New Session, when clicking it we import all results from our recent workspace

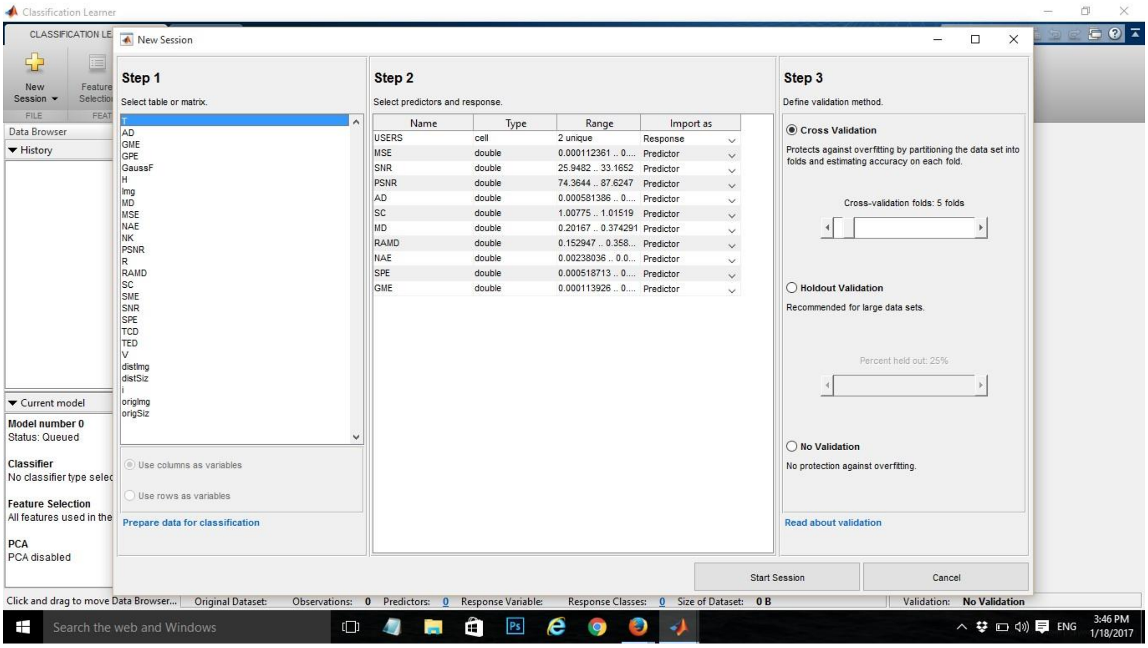


Figure 21: Select T that Refers to Table and Select our Response that is our Users, and the Predictors that are Refer to our Feature

Figure 21 the left side shows all the results imported from our work space including our table created T, we select our table, and the middle pane shows the selection of predictors and response then click start session for training process.

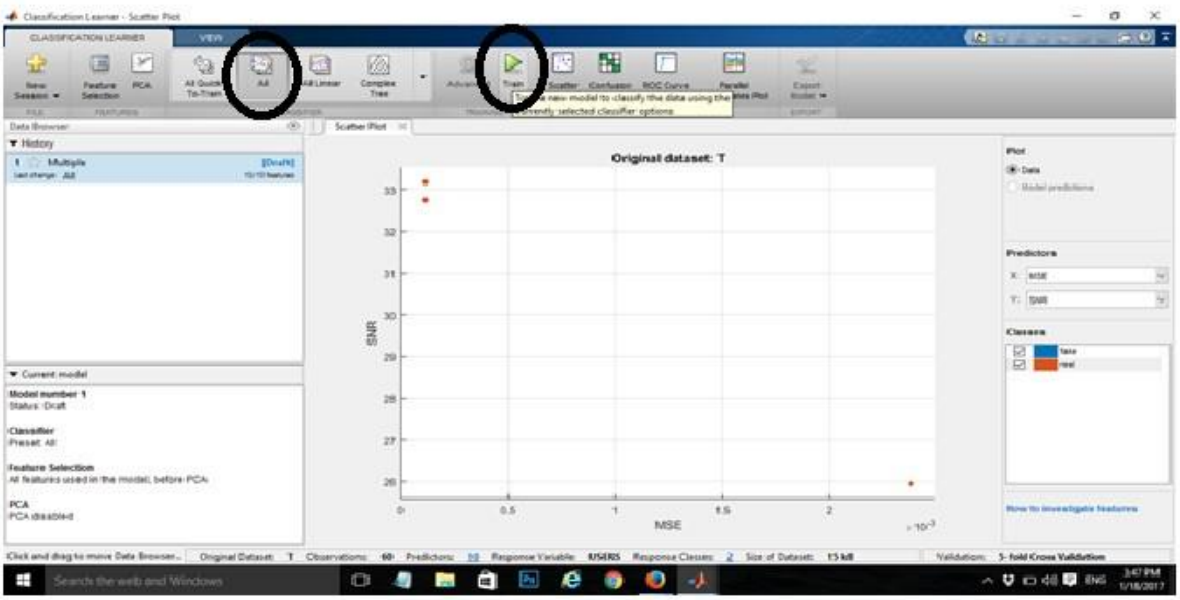


Figure 22: Select all for all Classifiers, Click Run to Run the Classification Process

Figure 22 shows all classifiers that can be selected for selecting all classifiers just click ALL classifiers and all classifiers will be imported and ready for run process, click run for starting the training process .

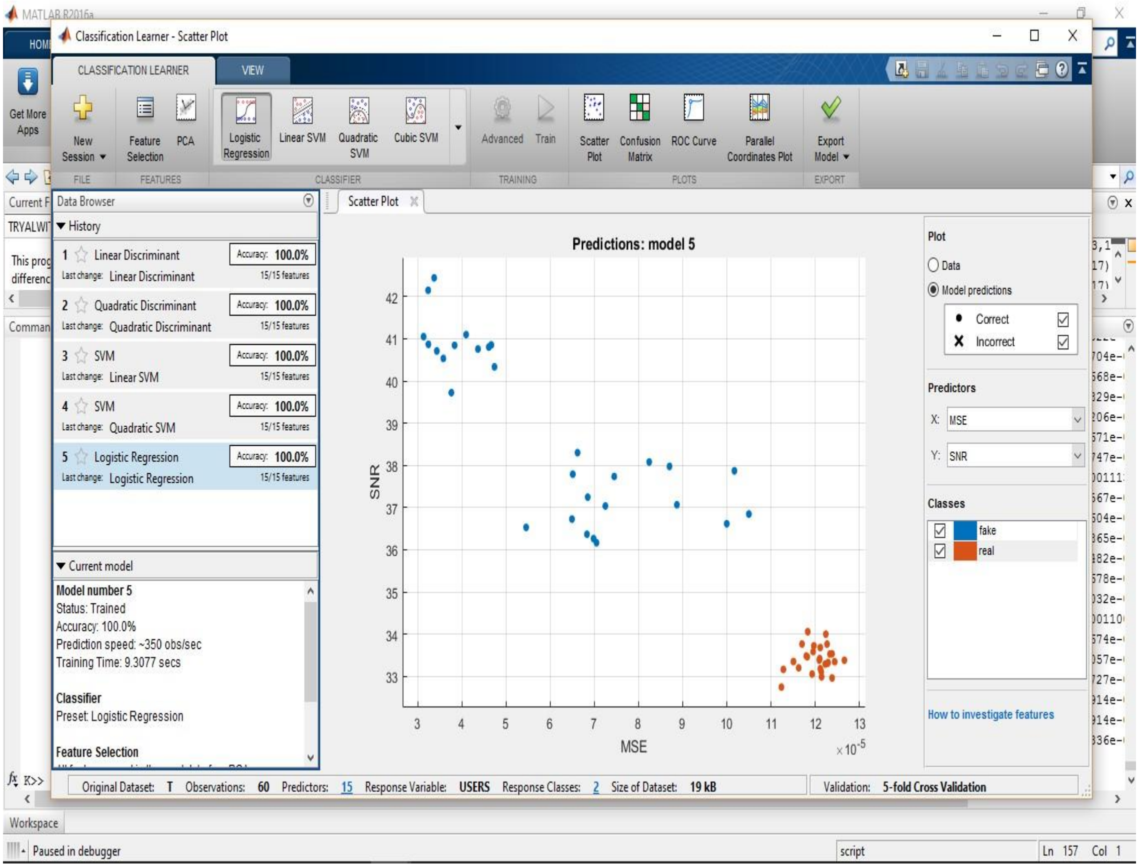


Figure 23: The Results of Training is Reported on the Left Side with (%), also we Can View our Results in Term of Scatter Plot, Confusion Matrix, ROC Curve, and Parallel Coordinate Plot, on the Top

Figure 23 shows the training reports and plots provided there are 4 types of classification plots provided by the classification learner application, scatter plot, confusion matrix, ROC curve, and parallel coordinates all these plots are provided with the present classification method that we choose at our left hand side, with percentage of accuracy presented.

The classification methods used are:

LDA

QDA

Linear SVM

Quadratic SVM

Logistic Regression

3.5 Implementation of Classification Subsystem

For classification process we need the model provided by training structure ten we input 4 images of different subjects from NUAA database for our system to classify if these images are fake or real.

Steps for implementing the classification process:

1) Export model of currently selected classifier to the work space for classification process
2) The function yfit = trainedClassifier.predictFcn(newT) must be added to your code for classification using the current model
3) Input 4 images for classification process
4) Run and classify

The below screenshots show how classification process works in details:

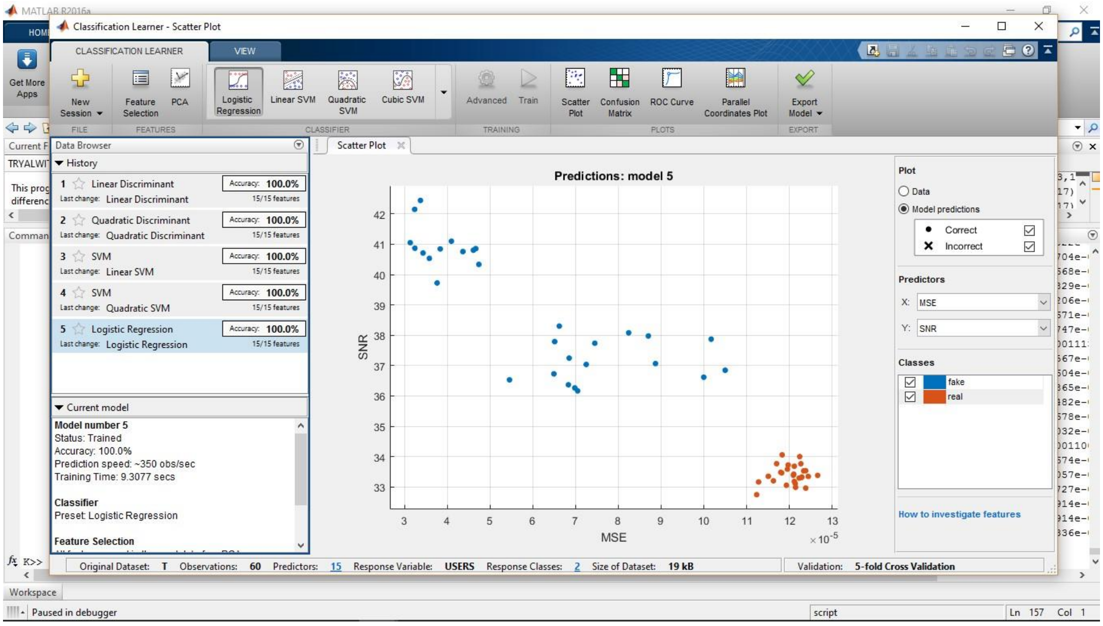


Figure 24: Training Model

Figure 24 shows the Training model we are going to export to our work space for classification process the left side shows the training information's of each classifier used the middle side shows the model of 60 real and fake training images.

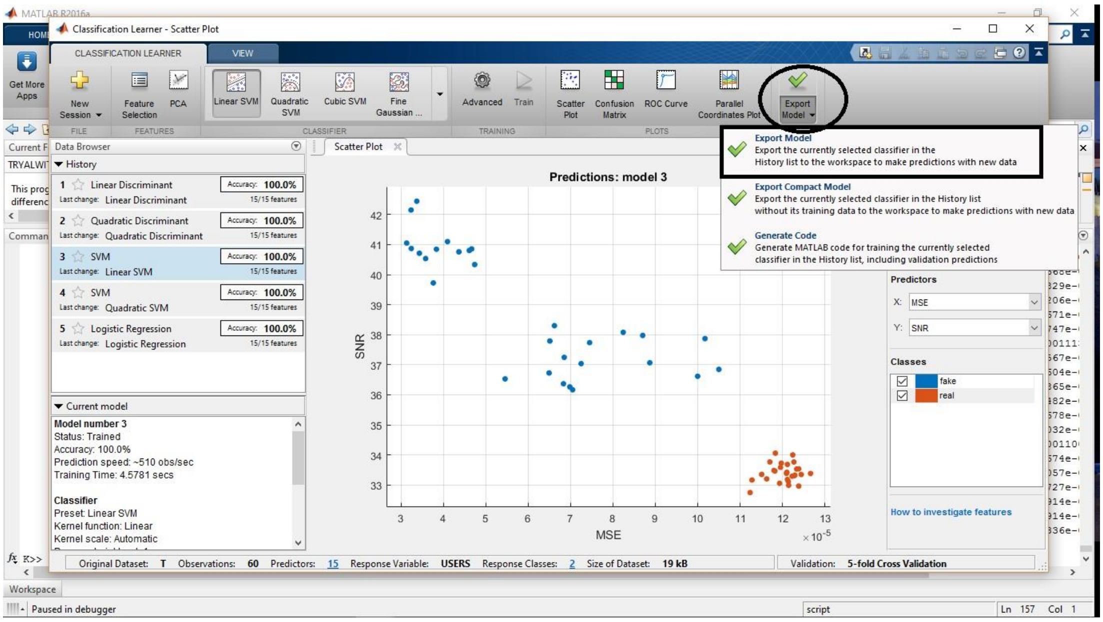


Figure 25: Exporting Process

Figure 25 shows after training our 60 real and fake images and obtaining a model we have to export the model to our work space for classification process to enter different images and let the model classify if they are real or fake, for exporting click on the upper right button Export model and select export the currently selected classifier in the history list to the work space to make predictions with new data, this will export the current model to the work space.

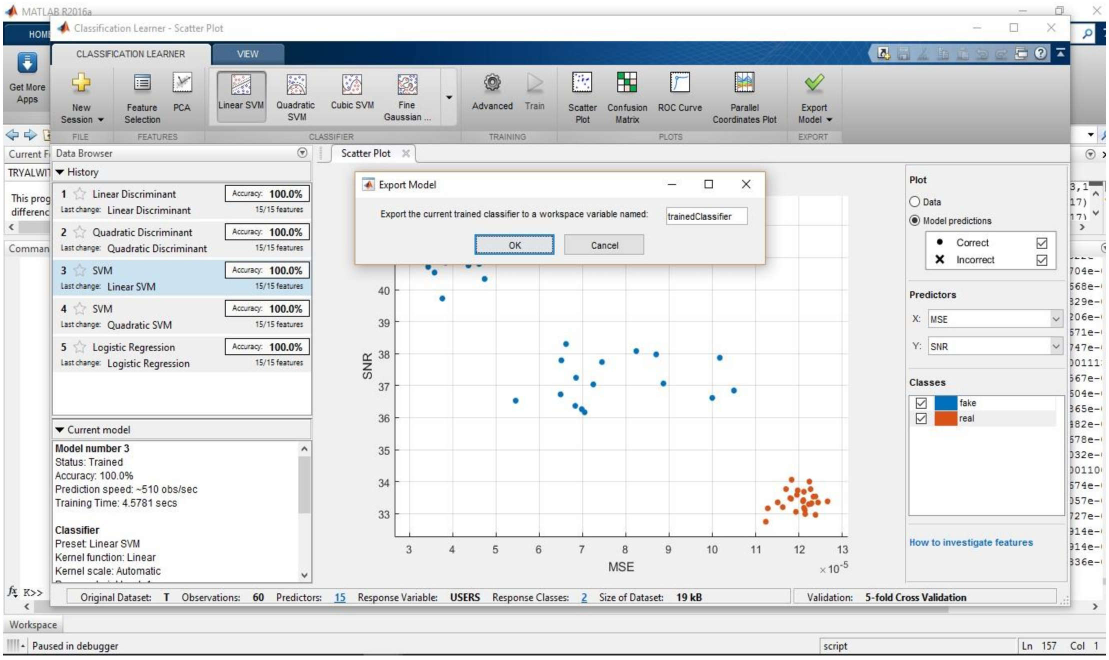


Figure 26: Name the Exporting Model

Figure 26 shows that we have to name our exported model because a line in the code will be added to our current code for classifying according to this model, type name and click ok, the name selected in the current implementation is trained classifier.

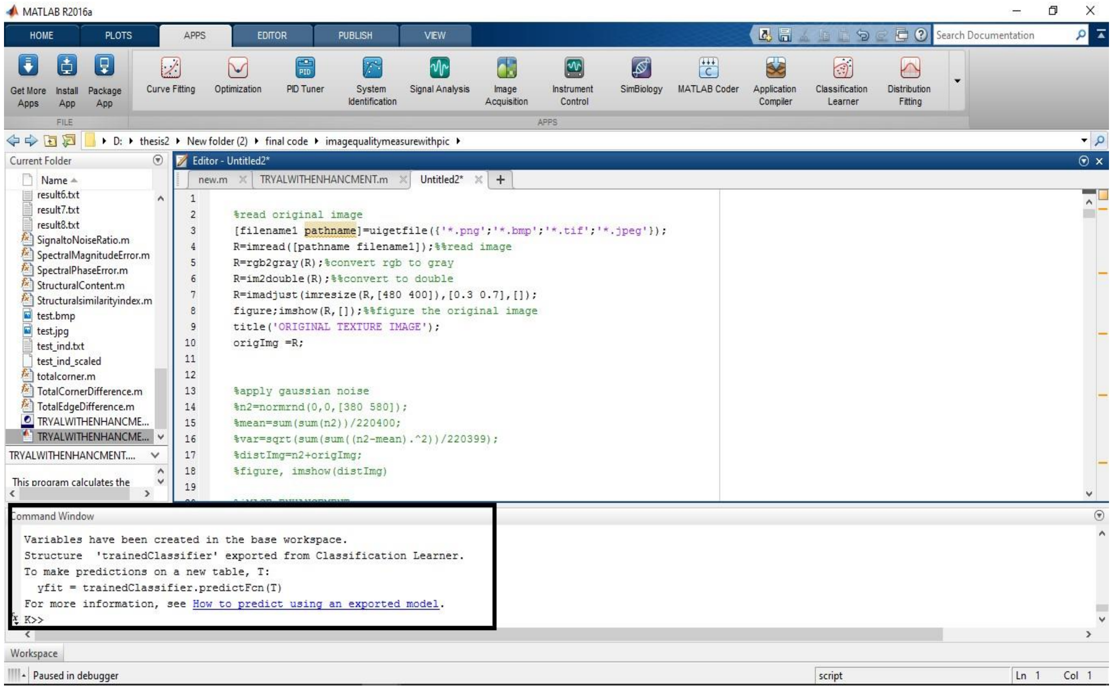


Figure 27: Exported Training Model

Figure 27 shows our exported training model in the work space for classification process it says that: Variables have been created in the base work space structure “trained classifier” exported from classification Learner. To make prediction on a new table T, code for classification process is form (line 6-81 Appendix A) in line 6 we can adjust the number of images we want to input from 60 to 4 for classification and in line 82 we have to add the following function to our code of input images for classifying according to the current training model: yfit = trainedclasifier.predictFcn(T) .

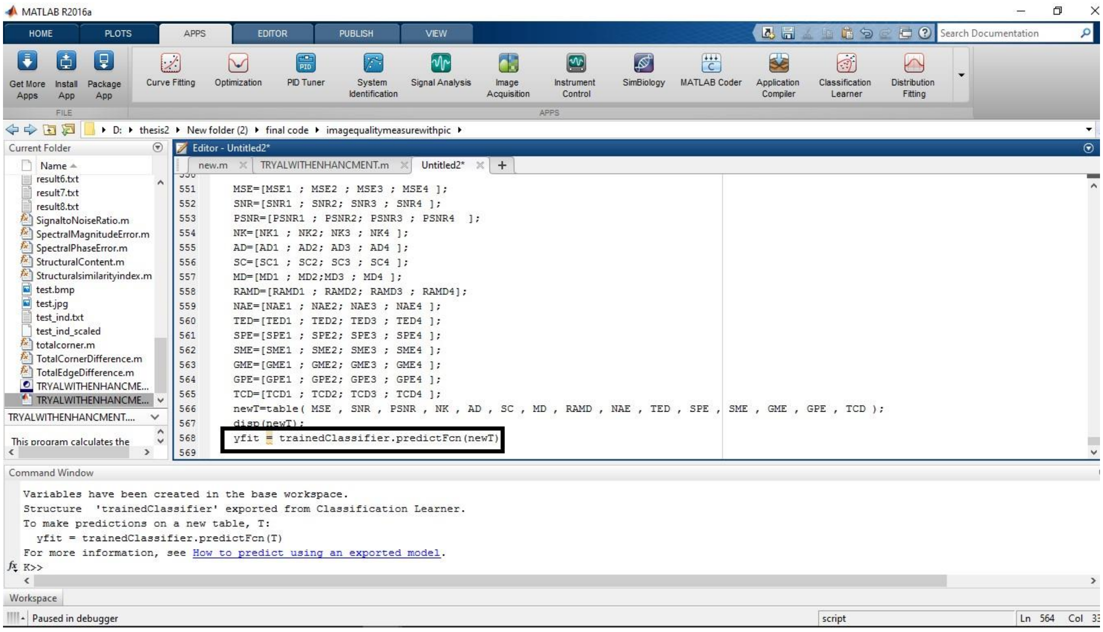


Figure 28: Code and Function for Classification

Figure 28 shows the code in the middle side it is not complete due to screenshot limitation but the function added is clear in the box indicated in the middle side, we enter four images for classification process and according to the exported model the classifier classifies if the 4 input images are real or fake.

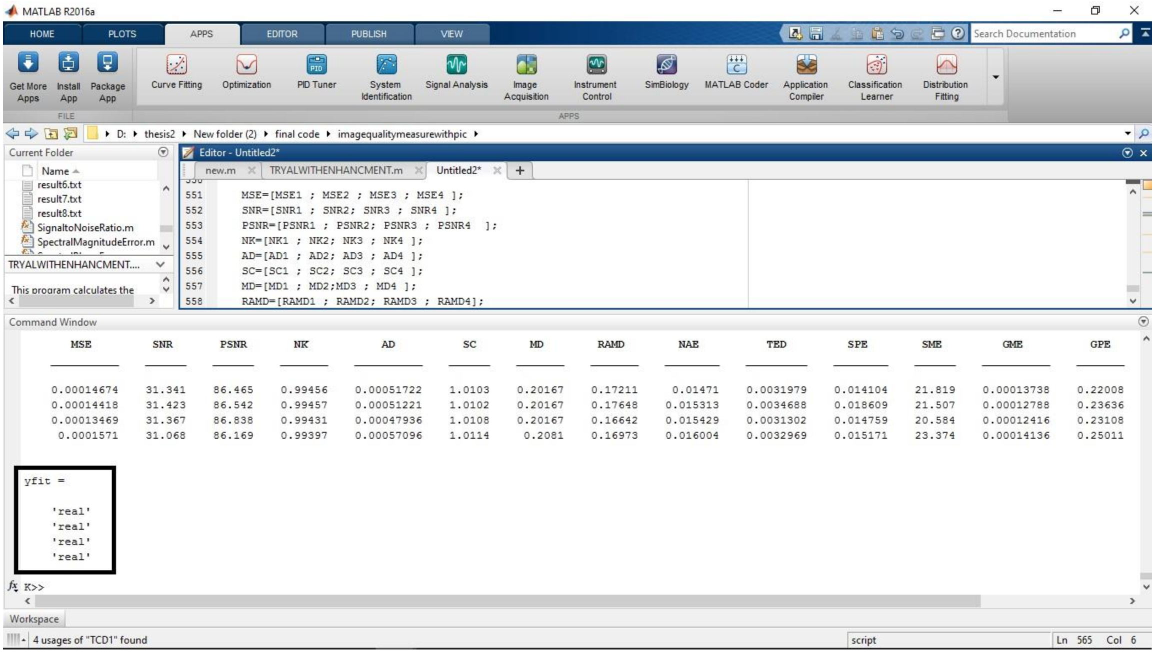


Figure 29: Table of Four Images for Classification

Figure 29 shows the table created after running our classification code the Table is not complete due to screenshot limitation. The Table contains each input image 15 quality features extracted and arranged in the corresponding Table without specifying which image is real and which is fake, According to the model exported from the classification learner application these images are classified either real or fake. For our current implementation we enter four real images of different users and they are classified correctly as real, as we can see in the box on work space yfit. (Figure 29 bottom left side)

3.6 Conclusion

In this chapter we show how we implemented our RFIDS. We implemented an overall structure having the following subsystems, Gaussian noise subsystem 3.2, feature extraction subsystem as seen in section 3.3 for 15 features MSE, SNR, PSNR, SC, MD, AD, NE, RAMD, NCC, TED, TCD, SPE, SME, GPE, GME, training as seen in section 3.4, and classification as seen in section 3.5. Each subsystem was implemented and tested in MATLAB 2016 see (Appendix A-C). These subsystems were implemented and tested in chapter 3 and code and screenshots are in Appendix C. Classifiers subsystem implemented in MATLAB 2016 as a separate application and exported into RFIDS (see Appendix D, We show how we export our training model to the main code for classification process). We also show how to name the exported model, and where the exported model appears in our classification structure.

Chapter 4

EXPERIMENTS ON RFIDS

This chapter shows the experiment setup for RFIDS and results of our experiments, the results are experimented on different datasets Tables 12 - 15 to ensure the presented methods quality, also we compared our proposed method with other state of-the-art methods Table 18 and see the efficiency of our work, the results are also conducted Table 16, 17 on different experiments on different types on quality features and show that with 15 features extracted for face images for training and same images for classification we get a good result.

4.1 Experiment Setup

Comparison between our experimental setup and the experimental setup used in [1]

Table 11: Comparison between RFIDS and IQA Based Method

RFIDS	IQA based method
Our measurement will be made on MATLAB R2016a the computer specifications where as follows 2.40 GHZ processor of 64-bit windows10-pc, with core i7, and 16 GB RAM memory,	The results were measured on a standard 64-bit Windows7-PC with a 3.4 GHz processor and 16 GB RAM memory, running MATLAB R2012b

Classification methods: LDA (linear discriminant analysis) QDA (Quadratic discriminant analysis) Linear SVM	Classification method: LDA (linear discriminant analysis)
Quadratic SVM Logistic Regression	
Results will be reported in terms of FAR (indicates the number of false samples which are identified as real), FFR (indicates the number of real samples considered as fake), and HTER=(FGR+FFR/2)	Results will be reported in terms of FAR (indicates the number of false samples which are identified as real), FFR (indicates the number of real samples considered as fake), and HTER=(FGR+FFR/2)
Best-5: SNR, PSNR, R-AMD, NAE, GME. Best-10: SNR, PSNR, R-AMD, NAE, GME, MSE, SPE, SC, AD, MD ALL	Best-5: NCC, RAMD, MAS, SPE, RRED Best-10: MSE, AD, SC, NCC, MD, RAMD, MAS, SME, SPE, MSE, PSNR, AD, SC, NCC, MD, SNR, RAMD, MAMS, SME, SPE, TCD, GME, VIF, NIQE Best-15: MSE, PSNR, AD, SC, NCC, MD, SNR, RAMD, MAMS, SME, SPE, TCD, GME, VIF, NIQE ALL
NUAA Photograph Imposter Database [39]	Replay-Attack database [40]

Table 11 the experimental setup of RFIDS and IQA base method are almost the same as there are slightly difference. The difference are IQA based method uses 25 image quality features where RFIDS uses 15 image quality features, the database used in IQM based method is Replay-Attack database [40], and in RFIDS we used NUAA Photograph Imposter Database [39], also the number of classifiers used are different as we used 5 classification methods namely LDA, QDA, Linear SVM, Quadratic SVM, and Logistic Regression, in IQA based method they used only LDA. Apart from this difference al other experimental setups are similar.

4.2 Code Explanation for Experiments Conducting

The first part of the code is the Gaussian filtering subsystem that consist of image input, converting image from rgb to grayscale, then converting the input image to double, and apply filtering using a 3*3 Gaussian filter, and resizing of original and distorted image to the same size see (Appendix A line 1-29).

The second part of the code is the feature extraction subsystem, in this part there are 15 features implemented each feature has an equation for calculation, the input to this subsystem are 2 images an original image and distorted image using this two images each function can successfully calculate the feature value. Then the 30 real face images and 30 fake face images together with their 15 image quality measures are combined in a vector and arranged in a table to be imported in the classification learner application for training process see section 3.4.

The third part is classification process using classification learner application, that classifies our images to real and fake images using 5 different classification methods

LDA, QDA, Linear SVM, Quadratic SVM, Logistic Regression for screenshots see Figure 24, 25, 26, 27, 28, 29.

4.3 Experimental Results Based on NUAA Database

The Experiments done are based on NUAA database using subject-4 for training and different subjects for classification.

Table 12: Training Results Using NUAA Database Subject-4

Subject number : 4, 60 face images (30 real/30 fake)				
Classifier	FFR (%)	FGR (%)	HTER (%)	Training time(sec)
LDA	0	0	0	1.5614
QDA	0	0	0	1.8197
Linear SVM	0	0	0	2.8546
Quadratic SVM	0	0	0	1.6242
Logistic Regression	0	0	0	7.0507

NUAA database subject-4 seen in section 2.5 is used to obtain the results in Table 12 was in terms of FFR equation (2.22), FGR equation (2.21), and HTER equation (2.23) from 4 classifiers on subject number 4 which is calculated using confusion matrix (Appendix C1, Figure C.2), a detailed Table C.1 that shows the result of 15 image quality measurements calculations with 30 real face images and 30 fake samples is provided. LDA is given from Figure C.1- C.3, QDA is given from Figure C.4- C.6, Linear SVM is given from Figure C.7- C.9, Quadratic SVM is given from Figure C.10- C.12, and Logistic Regression is given from Figure C.13- C.15.

Table 13: Classification Results Using NUAA Database

4 face images with different database subjects			
Classifier	FFR (%)	FGR (%)	HTER (%)
LDA	0	0	0
QDA	0	0	0
Linear SVM	0	0	0
Quadratic SVM	0	0	0
Logistic Regression	0	0	0

Table 13 shows the classification results using 4 images and 5 classifications for training models, the results were obtained by input of different types of images, fake and real, and the classifier successfully detected fake images (see appendix D for screenshots).

Table 14 results are obtained using 4 different images for classification and best-5 quality measures: SNR, PSNR, RAMD, NAE, GME.

Table 14: Results Obtained On NUAA Database Based On Best-5

Classifier	FFR (%)	FGR (%)	HTER (%)
LDA	10	0	5
QDA	11.6	1.6	6.6
Linear SVM	5	0	2.5

Quadratic SVM	3.3	0	1.6
Logistic Regression	3.3	0	1.6

Results obtained in Table 14 is in terms of FFR, FGR, and HTER from 4 classifiers on best-5, a detailed table that shows the result of 15 image quality measurements calculations with 4 images is provided together with screenshots of 5 different classifiers see (Appendix C.1 Table C.2). And best-5 was selected according to parallel distribution plot see (Appendix C.1, Figure C.3) we see that some features represent in more difference than others so we choose best features according to large difference of quality measures see Table C.4 it shows the minimum and maximum of each feature calculation.

Table 15 results are obtained using 4 different images for classification and best-10 quality measures: SNR, PSNR, RAMD, NAE, GME, MSE, SPE, SC, AD, MD.

Table 15: Results Obtained For Subject 4 Based On Best-10

Classifier	FFR (%)	FGR (%)	HTER (%)
LDA	6.6	1.6	4.1
QDA	3.3	1.6	2.45
Linear SVM	1.6	0	0.8
Quadratic SVM	3.3	0	1.6

Logistic Regression	8.3	1.6	4.95

Results obtained in Table 15 is in terms of FFR, FGR, and HTER from 4 classifiers on best-10, a detailed table that shows the result of 15 image quality measurements calculations with 4 images is provided together with screenshots of 5 different classifiers see (Appendix C1, Table C.3). And best-10 was selected according to parallel distribution plot see (Appendix C.1, Figure C3) we see that some features represent in more difference than others so we choose best features according to large difference of quality measures

From the results in Table 12 that are done on subject 4 NUAA database for training using 5 different classifiers LDA, QDA, Linear SVM, Quadratic SVM, Logistic Regression using our implemented code (see appendix A-P), we can see Table 4.3 and consider our proposed system as a comparative discriminator system when it comes to detecting false from real samples, Linear SVM is considered as the best discriminator when number of measures are decreased, based on LDA it gives the best execution time in all dataset experiments. On tables 14, 15 we presented the results conducted on different number of features to ensure the performance of the total 15 features, the experiments were conducted on database [39], using our implemented code [Appendix A-P].

Comparison between our RFIDS method and other state-of-art methods based on printed face note that our method uses linear SVM as a classifier:

Table 16: Comparison between RFIDS Method and Other State-of-Art Methods in Term of Spoofed Printed Faces.

Methods	FAR	FGR	HTER
IQA-based[1]:	0.0	1.0	0.5
AMILAB[32]	0.0	1.2	0.6
CASIA[32]	0.0	0.0	0.0
IDIAP[32]	0.0	0.0	0.0
SIANI[32]	0.0	21.2	10.6
UNICAMP[32]	1.2	0.0	0.6
UOULU[32]	0.0	0.0	0.0
RFIDS	0.0	0.0	0.0

Table 16 shows our proposed method RFIDS in comparison with other existing methods, our method is highly competitive in term of detecting fake faces and it shows HTER equal to 0, also some existing methods showed similar results but using different techniques, from this table we can say that using 15 image quality measures (parameters) our system showed better results than IQA based method that uses 25 image quality measures.

4.4 Conclusion

Chapter 4 shows experimental setup used in our proposed method and compared it to IQA based method that has similar setup except the database and number of image quality features, we show the results of our experiments done on NUAA database [39], and the results are trained using subject-4 Table 12 show a 0% of HTER in all classifiers. and classification results are present in table 13 shows the presented method quality using 5 different classification methods all results present HTER of 0%., we also show results based on best-5 and best-10 Table 14 and 15 and we introduced a method of how we selected this best called parallel distribution in (Appendix C, Table C.4) that shows the minimum and maximum of each feature based on 60 face images 30 real and 30 fake we can see that using Best-5 and Best-10 image quality features we can see that some error rates are present but we consider Liner SVM as our best classification method that give HTER of 0.8% in Best-10 features, also we compared our proposed method with other state-of-the-art methods in Table 16 to show our 0% result of HTER, three methods showed similar results of 0% HTER but using different techniques, based on image quality measures technique we used 15 features and showed result of 0% in HTER and IQA based method that uses 25 features showed higher HTER with 0.5%. Both methods used almost similar experimental setups.

Chapter 5

CONCLUSION

Recently biometric researches against spoofing attacks has been an important role of study, today we can examine the improvement of this biometric security technology against challenging methods such as spoofing attacks, we made a literature survey of similar methods, and saw that existing methods used different number of image quality features, and also present different types of classification methods, the results were tested on different databases, we can say that in recent years the results obtained on detecting fake faces were comparative but there was a clear absence of perfect results. Based on this investigation we defined problems of previous systems we implement RFIDS.

We show how we implemented our RFIDS, we implemented the overall structure using the following subsystems, Gaussian filtering subsystem, feature extraction subsystem, training mode and classification mode subsystems, each subsystem was implemented and tested in MATLAB2016.

We show experimental setup used in our proposed method and compared it to IQA based method that has similar setup except the database and number of image quality features. We show the results of our experiments done on NUAA database [39], and the results are trained using subject-4 which show a 0% of HTER in all classifiers. Classification results that are presented show the presented methods quality using 5 different classification methods. All results present HTER of 0% and we also show results based on best-5 and best-10. We introduced a method of how we selected the best according to parallel distribution that shows the minimum and maximum of each feature based on 60 face images 30 real and 30 fake images. We can see that using Best-5 and Best-10 image quality features, some error rates present but we consider Liner SVM as our best classification method that give HTER of 0.8% in Best-10 features. Also we compared our proposed method with other state-of-the-art methods to show our 0% result of HTER. Three methods showed similar results of 0% HTER but using different techniques. Based on image quality measures technique we used 15 features and showed result of 0% in HTER and IQA based method that uses 25 features showed higher HTER with 0.5%. Both methods used almost similar experimental setups.

Our future work will be aiming implementation of different types on biometric traits such as fingerprint, iris, etc… In order to conduct a multi-biometric system, we will include more classifiers to discriminate between real and fake images to ensure protection strategy.

REFERENCES

[1] Galbally, J., Marcel, S., & Fierrez, J. (2014). Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition. *IEEE transactions on image processing*, *23*(2), 710-724.

[2] Anjos, A., & Marcel, S. (2011, October). Counter-measures to photo attacks in face recognition: a public database and a baseline. In *Biometrics (IJCB), 2011 international joint conference on* (pp. 1-7). IEEE.

[3] Matsumoto, T. (2004)"Artificial irises: importance of vulnerability analysis." In *Proc. Asian Biometrics Workshop (AWB)*, vol. 45, no. 8.

[4] Galbally, J., Cappelli, R., Lumini, A., Gonzalez-de-Rivera, G., Maltoni, D., Fierrez, J. & Maio, D. (2010). An evaluation of direct attacks using fake fingers generated from ISO templates. *Pattern Recognition Letters*, *31*(8), 725-732.

[5] Hennebert, J., Loeffel, R., Humm, A., & Ingold, R. (2007, August). A new forgery scenario based on regaining dynamics of signature. In *International Conference on Biometrics* (pp. 366-375). Springer Berlin Heidelberg.

[6] Maltoni, D., Maio, D., Jain, A., & Prabhakar, S. (2009). *Handbook of fingerprint recognition*. Springer Science & Business Media.

[7] Cappelli, R., Maio, D., Lumini, A., & Maltoni, D. (2007). Fingerprint image reconstruction from standard templates. *IEEE transactions on pattern analysis and machine intelligence*, *29*(9).

[8] Shah, S., & Ross, A. (2006, October). Generating synthetic irises by feature agglomeration. In *Image Processing, 2006 IEEE International Conference on* (pp. 317-320). IEEE.

[9] Saranya, S., Sherline, S. V., & Maheswari, M. (2016, March). Fake biometric detection using image quality assessment: Application to iris, fingerprint recognition. In *Science Technology Engineering and Management (ICONSTEM), Second International Conference on* (pp. 98-103). IEEE.

[10] Sayood, K. (2002). Statistical evaluation of image quality measures. *Journal of Electronic imaging*, *11*(2), 206-223.

[11] Huynh-Thu, Q., & Ghanbari, M. (2008). Scope of validity of PSNR in image/video quality assessment. *Electronics letters*, *44*(13), 800-801.

[12] Yao, S., Lin, W., Ong, E., & Lu, Z. (2005, September). Contrast signal-to-noise ratio for image quality assessment. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on* (Vol. 1, pp. I-397). IEEE.

[13] Eskicioglu, A. M., & Fisher, P. S. (1995). Image quality measures and their performance. *IEEE Transactions on communications*, *43*(12), 2959-2965.

[14] Martini, M. G., Hewage, C. T., & Villarini, B. (2012). Image quality assessment based on edge preservation. *Signal Processing: Image Communication*, *27*(8), 875-882.

[15] Nill, N. B., & Bouzas, B. (1992). Objective image quality measure derived from digital image power spectra. *Optical engineering*, *31*(4), 813-825.

[16] Liu, A., Lin, W., & Narwaria, M. (2012). Image quality assessment based on gradient similarity. *IEEE Transactions on Image Processing*, *21*(4), 1500-1512.

[17] 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.

[18] *LIVE* [Online]. (2012), Retrieved from http://live.ece.utexas.edu/research/Quality/index.htm

[19] Sheikh, H. R., & Bovik, A. C. (2006). Image information and visual quality. *IEEE Transactions on image processing*, *15*(2), 430-444..

[20] Soundararajan, R., & Bovik, A. C. (2012). RRED indices: Reduced reference entropic differencing for image quality assessment. *IEEE Transactions on Image Processing*, *21*(2), 517-526.

[21] Wang, Z., Sheikh, H. R., & Bovik, A. C. (2002). No-reference perceptual quality assessment of JPEG compressed images. In *Image Processing. 2002. Proceedings. 2002 International Conference on* (Vol. 1, pp. I-I). IEEE..

[22] Zhu, X., & Milanfar, P. (2009, July). A no-reference sharpness metric sensitive to blur and noise. In *Quality of Multimedia Experience, 2009. QoMEx 2009. International Workshop on* (pp. 64-69). IEEE.

[23] 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.

[24] Mittal, A., Soundararajan, R., & Bovik, A. C. (2013). Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, *20*(3), 209-212.

[25] Fernandes, S. L., & Bala, G. J. (2016). Developing a Novel Technique for Face Liveness Detection. *Procedia Computer Science*, *78*, 241-247..

[26] Preethi.V and S.Chidambaram (2015). Fake Multi-biometric Detection for Applications of Fingerprint, Iris and Face Recognition, *International Journal of Engineering Trends and Technology (IJETT)* – Volume 21 Number 2 – March 2015

[27] Suresh, P. (2014). Find Pretend Biometric Mistreatment Image Quality Assessment for Animateness Detection. *Asian Journal of Technology & Management Research [ISSN: 2249–0892]*, *4*(02)..

[28] Costa-Pazo, A., Bhattacharjee, S., Vazquez-Fernandez, E., & Marcel, S. (2016, September). The REPLAY-MOBILE Face Presentation-Attack Database. In *Biometrics Special Interest Group (BIOSIG), 2016 International Conference of the* (pp. 1-7). IEEE.

[29] Vigneshwaran, S, M.Suresh, and Dr.R.Meenakumari. (2015). An SVM based Statistical Image Quality Assessment for Fake Biometric Detection. *International Journal for Trends in Engineering & Technology Volume 4 Issue 1 – April 2015.*

[30] INDUMATHI, S. S. M. R., & ME, M. (2015). Fake Biometric Detection Using Improved Features in Image and Dempster-Shafer Method. *signs*, *2*(3).

[31] Rinu Prakash T, Vipin Thomas. (2015). Biometric Security System for Fake Detection using image Quality Assessment Techniques. *International Journal on Recent and Innovation Trends in Computing and Communication Volume: 3 Issue: 9.*

[32] Chakka, M. M., Anjos, A., Marcel, S., Tronci, R., Muntoni, D., Fadda, G., ... & Roli, F. (2011, October). Competition on counter measures to 2-d facial spoofing attacks. In *Biometrics (IJCB), 2011 International Joint Conference on* (pp. 1-6). IEEE.

[33] Chingovska, I., Yang, J., Lei, Z., Yi, D., Li, S. Z., Kahm, O., ... & Komulainen, J. (2013, June). The 2nd competition on counter measures to 2D face spoofing attacks. In *Biometrics (ICB), 2013 International Conference on* (pp. 1-6). IEEE..

[33] Fontani, M., Bianchi, T., De Rosa, A., Piva, A., & Barni, M. (2011, November). A Dempster-Shafer framework for decision fusion in image forensics. In *Information Forensics and Security (WIFS), 2011 IEEE International Workshop on* (pp. 1-6). IEEE.

[34] INDUMATHI, S. S. M. R., & ME, M. (2015). Fake Biometric Detection Using Improved Features in Image and Dempster-Shafer Method. *signs*, *2*(3).

[35] Scatter Plot. (25 November 2016), Retrieved from https://en.wikipedia.org/wiki/Scatter_plot

[36] Confusion Matrix. (11 October 2016), Retrieved from https://en.wikipedia.org/wiki/Confusion_matrix

[37] ROC Curve. (4 January 2017), Retrieved from https://en.wikipedia.org/wiki/Reciever_operating_characteristics

[38] Parallel coordinate. (12 January 2017), Retrieved from https://en.wikipedia.org/wiki/Parallel_coordinates

[39] Tan, X., Li, Y., Liu, J., & Jiang, L. (2010, September). Face liveness detection from a single image with sparse low rank bilinear discriminative model. In *European Conference on Computer Vision* (pp. 504-517). Springer Berlin Heidelberg.

[40]Database Replay-Attack Retrieved from https://www.idiap.ch/dataset/replayattack

[41] Replay-Mobile Database Retrieved from https://www.idiap.ch/dataset/replay-mobile .

[42] Gaussian blur (11 January 2017), Retrieved from https://en.wikipedia.org/wiki/Gaussian_blur .

[43] False acceptance rate Retrieved from http://www.webopedia.com/TERM/F/false_acceptance.html .

[44] Gunasekar, S., Ghosh, J., & Bovik, A. C. (2014). Face detection on distorted images augmented by perceptual quality-aware features. *IEEE transactions on information forensics and security*, *9*(12), 2119-2131.

APPENDICES

Appendix A: Main Code

```
%This program calculates the difference Image/Picture Quality Measures

%Clear Memory & Command Window

1. clc;
2. clear all;
3. close all;
4. char origImg;
5. char distImg;
6. for i=1:60;
%read original image
7. [filename1 pathname]=uigetfile({'*.png';'*.bmp';'*.tif';'*.jpeg'});
8. R=imread([pathname filename1]);%%read image
9. R=rgb2gray(R);%convert rgb to gray
10. R=im2double(R);%%convert to double
11. R=imadjust(imresize(R,[480 400]),[0.3 0.7],[]);
12. figure;imshow(R,[]);%%figure the original image
13. title('ORIGINAL TEXTURE IMAGE');
14. realImg =R;


%apply gaussian noise

%n2=normrnd(0,0,[380 580]);
```

```
%mean=sum(sum(n2))/220400;

%var=sqrt(sum(sum((n2-mean).^2))/220399);

%distImg=n2+origImg;

%figure, imshow(distImg)


%iMAGE ENHANCEMENT


%Gaussian filter using MATLAB built_in function

%Read an Image

16.  Img = R;

17.  H = fspecial('Gaussian',[3 3],0.5);

18.  GaussF = imfilter(R,H);

19.  figure,imshow(GaussF);

20.  enhImg=GaussF;


%convert images to double

21.  enhImg=im2double(ehnImg);

22.  realImg=im2double(realImg);


%Size Validation

23.  realImg = size(realImg);

24.  distSiz = size(enhImg);

25.  sizErr = isequal(realImg, distSiz);
```

```
if(sizErr == 0)
disp('Error: Original Image & Distorted Image should be of same dimensions');
return;
end
%Mean Square Error
MSE = MeanSquareError(realImg, enhImg);
disp('1-Mean Square Error = ');
disp(MSE);

%spectral magnitude error
SME = SpectralMagnitudeError(realImg, enhImg);
disp('2-SME = ');
disp(SME);

%spectral phase error
SPE = SpectralPhaseError(realImg, enhImg);
disp('3-SPE = ');
disp(SPE);

%gradient magnitude error
GME = GradientMagnitudeError(realImg, enhImg);
disp('4-GME = ');
disp(GME);
```

```
%gradient phase error
GPE = GradientphaseError(realImg, enhImg);
disp('5-GPE = ');
disp(GPE);

%signal to noise ratio
SNR = SignaltoNoiseRatio(realImg, enhImg);
disp('6-SNR = ');
disp(SNR);

%Peak Signal to Noise Ratio
PSNR = PeakSignaltoNoiseRatio(realImg, enhImg);
disp('7-PSNR = ');
disp(PSNR);

%Normalized Cross-Correlation
NK = NormalizedCrossCorrelation(realImg, enhImg);
disp('8-NCC  = ');
disp(NK);

%Average Difference
AD = AverageDifference(realImg, enhImg);
disp('9-AD  = ');
disp(AD);
```

```
%Structural Content

56. SC = StructuralContent(realImg, enhImg);
57. disp('10-SC  = ');
58. disp(SC);
%Maximum Difference

59. MD = MaximumDifference(realImg, enhImg);
60. disp('11-MD = ');
61. disp(MD);



%RaverageMD

62. RAMD = RarverageMD(realImg, enhImg);
63. disp('12-RAMD = ');
64. disp(RAMD);


%Normalized Absolute Error

65. NAE = NormalizedAbsoluteError(realImg, enhImg);
66. disp('13-NAE = ');
67. disp(NAE);


%Total Edge Difference

68. TED = TotalEdgeDifference(realImg, enhImg);
69. disp('14-TED  = ');
70. disp(TED);
```

%TotalCornerDifference

71. TCD = TotalCornerDifference(realImg, enhImg);

72. disp('15-TCD = ');

73. disp(TCD);

%vector of original image features

74. V(:,i)=[MSE; SNR; PSNR; NK; AD; SC; MD; RAMD; NAE; TED; SPE; SME;

GME; GPE ;TCD];

75. disp('V1=');

76. disp(V(:,i));

77. end

%create table for classification

78. USERS={'real'; 'fake'};

79. T=table(USERS, MSE , SNR , PSNR , AD , SC , MD , RAMD , NAE , SPE,

GME, SME, GPE, TCD, TED, NCC);

80. T(1: 60 , :)

81. disp(T);% create table using real and fake users and 15 image quality features.

Appendix B: Code of Feature Extraction Subsystem

B1: Mean Squared Error Function

%Program for Mean Square Error Calculation

```
function MSE = MeanSquareError(realImg, ehnImg)
realImg = double(realImg);
ehnImg = double(ehnImg);
[M N] = size(realImg);
error = realImg - ehnImg;
MSE = sum(sum(error .* error)) / (M * N);
```

B2: Peak Signal To Noise Ratio Function

```
%Program for Peak Signal to Noise Ratio Calculation

function PSNR = PeakSignaltoNoiseRatio(realImg, ehnImg)
realImg = double(realImg);
ehnImg = double(ehnImg);
[M N] = size(realImg);
error = realImg - ehnImg;
MSE = sum(sum(error .* error)) / (M * N); if(MSE > 0)
PSNR = 10*log10((255.*255)/MSE); else     PSNR = 99; end
```

B3: Signal To Noise Ratio Function

```
function SNR = SignaltoNoiseRatio(realImg, ehnImg)
realImg = double(realImg);
ehnImg = double(ehnImg);
```

```
[M, N] = size(realImg);
error = realImg - ehnImg;
MSE = sum(sum(error .* error)) / (M * N);

if(MSE > 0)
SNR = 10*log10((sum(sum(realImg.*realImg)))./(M .* N .* MSE));
    else    SNR = 99; end
```

B4: Structural Content Function

```
%Program for Structural Content Calculation
function SC = StructuralContent(realImg, ehnImg)
realImg = double(realImg);
ehnImg = double(ehnImg);
SC = sum(sum(realImg .* realImg)) ./ sum(sum(ehnImg .* ehnImg)
```

B5: Maximum Difference Function

```
%Program for Maximum Difference Calculation
function MD = MaximumDifference(realImg, ehnImg)
realImg = double(realImg);
ehnImg = double(ehnImg);
error = realImg - ehnImg;
MD = max(max(abs(error)));
```

B6: Average Difference Function

%Program for Average Difference Calculation

```
function AD = AverageDifference(realImg, ehnImg)
realImg = double(realImg);
ehnImg = double(ehnImg);
[M N] = size(realImg);
error = realImg - ehnImg;
AD = sum(sum(error)) / (M * N);
```

B7: Normalized Absolute Error Function

%Program for Normalized Absolute Error Calculation

```
function NAE = NormalizedAbsoluteError(realImg, ehnImg)
realImg = double(realImg);
ehnImg = double(ehnImg);
error = realImg - ehnImg;
NAE = sum(sum(abs(error))) ./ sum(sum(abs(realImg)));
```

B8: R-Averaged MD Function

```
function RAMD = RarverageMD(realImg, ehnImg)
realImg = double(realImg);
ehnImg = double(ehnImg);
error = realImg - ehnImg;
```

```
5. error=abs(error);

6. a=error(:);%convert matrix to vector

   c=flipud(unique(sort(a)));

7. resultat=c(1:10);        %top ten

% because there equals number , there is more than 10 classed. if you want take only

%10     include     this     code

resultat1=resultat(1:10,:)

R = 10;

8. RAMD = sum((abs(resultat1)))/R;
```

B9: Normalized cross correlation function

```
%Program for Normalized Cross Correlation Calculation

1. function NK = NormalizedCrossCorrelation(realImg, ehnImg)

2. realImg = double(realImg);

3. ehnImg = double(ehnImg);

4. NK = sum(sum(realImg .* ehnImg)) ./ sum(sum(realImg .*
   realImg));
```

B10: Total Edge Difference Function

```
1.  function TED = TotalEdgeDifference(realImg, ehnImg)

2.  realImg=edge(realImg,'sobel');

3.  figure, imshow(realImg)

4.  realImg = double(realImg);

5.  ehnImg=edge(ehnImg,'sobel');
```

```
6.  figure, imshow(realImg)
7.  ehnImg = double(ehnImg);
8.  [M, N] = size(realImg);
9.  error = realImg - ehnImg;
10. TED = sum(sum(abs(error))) / (M * N);
```

B11: Gradient Magnitude Error Function

```
%Program for GradientMagnitudeError Calculation

1. function GME = GradientMagnitudeError(realImg, enhImg)
2. realImg = double(realImg);
3. enhImg = double(enhImg);
% Gradient transform of image

4. [FX1,FY1] = gradient(realImg);
Gmagorg=sqrt((FX1.*FX1)+(FY1.*FY1));

[FX1,FY1] = gradient(enhImg);

Gmagdist=sqrt((FX1.*FX1)+(FY1.*FY1));

5. [M , N] = size(realImg);
6. error=abs(Gmagorg)-abs(Gmagdist); error=error.*error;
7. GME =sum(sum( error)) / (M * N)
```

B12: Gradient Phase Error Function

```
%Program for GradientMagnitudeError Calculation

function GPE = GradientphaseError(realImg, enhImg)
realImg = double(realImg);
enhImg = double(enhImg);
% Gradient transform of image

[FX1,FY1] = gradient(realImg);%gradient of originalimage
z=complex(FX1,FY1);%transfer to complex number
argGmagorg = angle(z);%angle of complex original image
[FX1,FY1] = gradient(enhImg);%gradient of ditortedimage
y=complex(FX1,FY1);%transfer to complex number
argGmadist = angle(y);%angle of complex distorted image
[M , N] = size(realImg);
error=abs(argGmagorg)-abs(argGmadist);
GPE =sum(sum( error.*error)) / (M * N
```

B13: Spectral Magnitude Error Function

```
%Program for SpectralMagnitudeError Calculation

function SME = SpectralMagnitudeError(realImg, enhImg)
realImg = double(realImg);
enhImg = double(enhImg);
```

```
% Fastfourier transform of image

4.  fftA = fft2(double(realImg));

5.  z1r=real(fftA);%image real part

6.  z1i=imag(fftA);%image imaginary part

7.  fftA1=sqrt((z1r.*z1r)+(z1i.*z1i));%gradient of image fftB =
    fft2(double(enhImg));

8.  z2r=real(fftB);%image real part

9.  z2i=imag(fftB);%image imaginary part

10. fftB1=sqrt((z2r.*z2r)+(z2i.*z2i));%gradient of image [M , N] =
    size(realImg); error=abs(fftA1)-abs(fftB1);

11. SME =sum(sum( error.*error)) / (M * N);
```

B14: Spectral Phase Error Function

```
%Program for SpectralPhaseError Calculation

1. function SPE = SpectralPhaseError(realImg, enhImg)

2. realImg = double(realImg);

3. enhImg = double(enhImg);

% Fastfourier transform of image 4. fftA =
fft2(double(realImg)); argfftA =angle(fftA); fftB =
fft2(double(enhImg)); argfftB =angle(fftB); [M , N] =
size(realImg); error=abs(argfftA)-abs(argfftB);
```

```
5. SPE =sum(sum( error.*error)) / (M * N);
```

B15: Total Corner Difference Function

```
%Program for TotalCornerDifference Calculation

1. function TCD = TotalCornerDifference(realImg, ehnImg)
2. realImg = double(realImg);
3. ehnImg = double(ehnImg);
4. NCRorig = detectHarrisFeatures(realImg); imshow(realImg);  hold on;
5. plot(NCRorig);
6. NCRorig = length(NCRorig);

7. NCRdist = detectHarrisFeatures(ehnImg); imshow(ehnImg);  hold on;
8. plot(NCRdist);
9. NCRdist = length(NCRdist);
10. max1= max(NCRorig,NCRdist);
11. TCD =(abs(NCRorig-NCRdist))/max1;
```

Appendix C: Screenshots of Training Results Obtained in [4.2]

The following are the screen shots of 4 different datasets, each dataset contains 60 images with 30 real and 30 fake face samples, and each face with 15 image quality assessments calculated and results provided. 5 different classifiers are calculated with different types of plots, each classifier is shown in terms of scatter plot, confusion matrix, ROC curve, and parallel coordinates.

Appendix C1

NUAA database subject-4:

This table shows the result of 15 image quality measurements calculations with 30 real face images and 30 fake samples for training.

Subject-4 60 face image samples with 15 (IQA) measures results:

USERS	MSE	SNR	PSNR	NK	AD	SC	MD	RAMD	NAE	TED	SPE	SME	GME	GPE	TCD
'real'	0.00013839	32.934	86.72	0.99604	0.00061522	1.0075	0.19492	0.17396	0.0092871	0.002625	0.012271	19.643	0.00012298	0.18871	0.043716
'real'	0.00014198	33.179	86.608	0.99625	0.00062833	1.0071	0.19256	0.16744	0.0088657	0.0027917	0.01416	20.009	0.00012931	0.19232	0.045198
'real'	0.00016147	32.598	86.05	0.99607	0.00061549	1.0074	0.20097	0.18342	0.0099305	0.00325	0.013656	24.756	0.00014512	0.20561	0.021858
'real'	0.00015828	32.435	86.137	0.9959	0.00062885	1.0077	0.21041	0.18983	0.010396	0.0031979	0.013765	24.115	0.00014209	0.20387	0.027273
'real'	0.00014756	32.735	86.441	0.99602	0.0006144	1.0075	0.20238	0.1655	0.010352	0.0034323	0.016672	21.714	0.00013139	0.20233	0.026756
'real'	0.000159	32.602	86.117	0.99601	0.00062285	1.0075	0.20167	0.18023	0.010156	0.0033542	0.013769	23.432	0.00014325	0.20296	0.032787
'real'	0.00017017	31.738	85.822	0.99535	0.00060042	1.0087	0.21421	0.1912	0.011653	0.0030885	0.01303	26.408	0.00015127	0.20376	0.032922
'real'	0.0001985	32.941	85.153	0.99635	0.00073598	1.0068	0.19495	0.18086	0.0089552	0.0029792	0.012547	29.901	0.00017819	0.23344	0
'real'	0.0001894	33.528	85.357	0.99685	0.00073604	1.0059	0.20786	0.18541	0.0080029	0.0030938	0.01147	28.721	0.0001749	0.21364	0.025
'real'	0.00019291	33.415	85.277	0.99676	0.00075341	1.0061	0.1908	0.17782	0.008108	0.0031458	0.0095362	27.349	0.00017559	0.21702	0.012146
'real'	0.00016202	33.986	86.035	0.99705	0.00064447	1.0055	0.18593	0.16199	0.0076339	0.0028906	0.013223	22.811	0.00014458	0.21233	0.053333
'real'	0.00018424	33.738	85.477	0.9969	0.00076227	1.0058	0.23706	0.18761	0.0076722	0.0026875	0.011457	25.557	0.00016804	0.21213	0.13187
'real'	0.00018953	33.754	85.354	0.99693	0.00076952	1.0058	0.23747	0.18922	0.0075694	0.0030573	0.010764	26.667	0.00017476	0.22635	0.05042
'real'	0.00019154	33.727	85.308	0.99694	0.00076266	1.0057	0.20439	0.18927	0.0076949	0.0029896	0.0085035	26.882	0.00017533	0.2182	0.04386
'real'	0.00019388	33.609	85.256	0.99688	0.00076477	1.0058	0.264	0.19447	0.0077714	0.0029635	0.0093597	27.578	0.00017721	0.21394	0.045872
'real'	0.00019459	33.693	85.24	0.99697	0.00074878	1.0057	0.2004	0.18591	0.0076079	0.003125	0.012308	28.398	0.00017551	0.22521	0.047619
'real'	0.00019479	33.583	85.235	0.99685	0.00077301	1.0059	0.22134	0.18637	0.0076587	0.002901	0.012273	27.396	0.00017778	0.21839	0.028846
'real'	0.00019392	33.671	85.255	0.99694	0.00074319	1.0057	0.1997	0.17885	0.0076774	0.0031979	0.012164	28.409	0.00017545	0.21967	0.032258
'real'	0.00019314	33.638	85.272	0.99688	0.00076306	1.0058	0.21931	0.18345	0.0076866	0.003099	0.0092178	27.348	0.00017547	0.2188	0
'real'	0.00019131	33.681	85.313	0.99692	0.00075223	1.0058	0.19907	0.17749	0.0076312	0.0029792	0.0098657	27.154	0.00017538	0.21423	0.039801
'real'	0.00019322	33.416	85.27	0.99674	0.00074967	1.0061	0.22075	0.19771	0.0080081	0.003	0.011538	27.671	0.00017647	0.20993	0.027174
'real'	0.00019588	33.393	85.211	0.99673	0.00075814	1.0061	0.23827	0.19304	0.0079388	0.002849	0.010265	28.007	0.00018014	0.21319	0.043478
'real'	0.00019382	33.647	85.257	0.99691	0.00074055	1.0058	0.20841	0.19325	0.0076741	0.00275	0.0078554	27.144	0.00017488	0.22058	0.025105
'real'	0.00019228	33.617	85.291	0.99686	0.00076428	1.0059	0.21396	0.18138	0.0077732	0.0027865	0.0098173	27.175	0.00017472	0.20894	0.018957
'real'	0.0001732	33.756	85.745	0.99692	0.00071376	1.0058	0.22671	0.17936	0.0080104	0.0037448	0.0152	26.932	0.00015593	0.19769	0.02
'real'	0.00017533	33.762	85.692	0.99696	0.00070509	1.0057	0.20167	0.17326	0.0079211	0.0037344	0.015011	26.664	0.00015638	0.19916	0.0076046
'real'	0.00017001	33.695	85.826	0.99686	0.00070015	1.0059	0.20167	0.17841	0.0081397	0.0034531	0.016233	25.693	0.0001493	0.19425	0.087379
'real'	0.00017006	33.572	85.825	0.99678	0.00070483	1.006	0.21368	0.19082	0.0081743	0.003526	0.015239	25.579	0.00015093	0.19677	0.00369
'real'	0.0001754	33.878	85.69	0.99701	0.00070713	1.0056	0.20167	0.18654	0.0077468	0.0033958	0.017633	26.602	0.0001549	0.20338	0.039568
'real'	0.00017986	33.745	85.581	0.99698	0.00070018	1.0056	0.19386	0.16788	0.0079178	0.0032135	0.015126	27.527	0.00015767	0.20057	0.011811
'fake'	4.8054e-06	38.497	101.31	0.99811	5.1049e-05	1.0037	0.12605	0.084648	0.0056693	0.0026615	0.069974	0.67508	3.9738e-06	0.088608	0.011111
'fake'	2.7936e-05	35.907	93.669	0.99724	0.00025209	1.0053	0.1785	0.12786	0.0051504	0.0032917	0.065515	3.0404	2.641e-05	0.10046	0.08
'fake'	5.6548e-05	38.105	90.607	0.99832	0.00036697	1.0032	0.15674	0.12984	0.0048454	0.0019219	0.021402	8.2467	5.0944e-05	0.17464	0.10078
'fake'	6.7866e-05	37.902	89.814	0.99829	0.00044861	1.0033	0.18731	0.15286	0.0042187	0.0023437	0.012336	8.639	6.6474e-05	0.17193	0.12903
'fake'	7.3403e-05	37.704	89.474	0.99823	0.00042351	1.0034	0.14932	0.1192	0.0045607	0.001599	0.010396	10.262	6.8378e-05	0.17164	0.060345
'fake'	7.9749e-05	37.393	89.114	0.99811	0.00043581	1.0036	0.15527	0.12049	0.0050042	0.0018333	0.007759	11.721	7.4845e-05	0.167	0.046296
'fake'	7.6885e-05	37.575	89.272	0.9982	0.00045617	1.0034	0.15878	0.12016	0.0048253	0.0020677	0.010794	11.792	6.581e-05	0.16944	0.030612
'fake'	4.8946e-05	40.346	91.234	0.99884	0.00051088	1.0022	0.14299	0.11298	0.002892	0.0027135	0.014089	5.3965	4.6254e-05	0.17298	0.18405
'fake'	6.2492e-05	39.865	90.173	0.99891	0.0004289	1.0021	0.14031	0.11609	0.0030518	0.0017969	0.009197	8.3214	5.5334e-05	0.15688	0
'fake'	7.0479e-05	37.927	89.65	0.99832	0.00038201	1.0032	0.14468	0.11537	0.0049119	0.0016667	0.0061615	10.824	6.0384e-05	0.17484	0.082645
'fake'	7.2001e-05	37.949	89.557	0.99834	0.00043616	1.0032	0.15264	0.11883	0.0044933	0.0020938	0.010704	10.933	6.5327e-05	0.16981	0.068966
'fake'	6.8657e-05	38.243	89.764	0.9984	0.00039086	1.0031	0.15815	0.12542	0.0039882	0.0019479	0.013604	10.08	6.548e-05	0.16881	0.033708
'fake'	7.2132e-05	38.102	89.549	0.99841	0.00037659	1.003	0.14464	0.11568	0.0042557	0.0020729	0.0087943	10.588	6.6047e-05	0.15915	0.03
'fake'	5.8876e-05	39.001	90.431	0.99865	0.00037691	1.0026	0.13285	0.10772	0.0039503	0.0024219	0.007488	8.7381	5.1857e-05	0.17351	0.10638
'fake'	5.3248e-05	39.356	90.868	0.99869	0.00035889	1.0025	0.12794	0.10517	5.3248e-05	0.0019167	0.0075881	7.4996	4.8106e-05	0.16136	0.11594
'fake'	5.8339e-05	39.064	90.471	0.99861	0.00038632	1.0027	0.13341	0.11713	5.8339e-05	0.0028073	0.013207	7.6931	5.3788e-05	0.16167	0.083333
'fake'	5.4537e-05	39.314	90.764	0.99869	0.00037334	1.0025	0.12617	0.11049	5.4537e-05	0.0024896	0.012009	7.7227	5.0296e-05	0.16755	0.042857
'fake'	5.0727e-05	39.599	91.078	0.99875	0.0003539	1.0024	0.13325	0.10884	5.0727e-05	0.0029635	0.0085575	7.179	4.7378e-05	0.16735	0.1125
'fake'	6.9204e-05	38.156	89.729	0.99839	0.00046398	1.0031	0.14299	0.1226	6.9204e-05	0.0027083	0.0087255	8.7369	6.6564e-05	0.16388	0.15584
'fake'	6.9204e-05	38.156	89.729	0.99839	0.00046398	1.0031	0.14299	0.1226	6.9204e-05	0.0027083	0.0087255	8.7369	6.6564e-05	0.16388	0.15584
'fake'	7.0132e-05	38.28	89.672	0.99841	0.00049164	1.003	0.13073	0.11275	7.0132e-05	0.001974	0.013808	8.3524	6.6919e-05	0.16484	0.045977
'fake'	7.2012e-05	38.132	89.557	0.9984	0.00041939	1.0031	0.13733	0.11587	7.2012e-05	0.001974	0.010302	10.322	6.7932e-05	0.15787	0.058252
'fake'	7.3692e-05	37.986	89.457	0.99834	0.00045478	1.0032	0.15024	0.11566	7.3692e-05	0.0015885	0.0087009	10.345	6.7358e-05	0.15682	0.1028
'fake'	6.26e-05	38.653	90.165	0.99848	0.00049749	1.0029	0.14915	0.11746	6.26e-05	0.0031406	0.017328	7.8429	5.9835e-05	0.16192	0.1358
'fake'	7.2664e-05	37.869	89.518	0.99832	0.00042005	1.0032	0.14159	0.11788	7.2664e-05	0.0020521	0.013308	9.3421	6.6064e-05	0.16855	0.068966
'fake'	7.0068e-05	37.929	89.676	0.99833	0.00040482	1.0032	0.1532	0.11865	7.0068e-05	0.0015833	0.010553	10.104	5.9415e-05	0.16326	0.10811
'fake'	6.0807e-05	38.755	90.291	0.99856	0.00044488	1.0028	0.14505	0.11464	6.0807e-05	0.0032448	0.011524	7.7638	5.3266e-05	0.15407	0.10843
'fake'	7.2794e-05	38.066	89.51	0.99836	0.00045375	1.0031	0.15034	0.12037	7.2794e-05	0.001901	0.012951	10.228	6.8728e-05	0.16456	0.13889
'fake'	5.2269e-05	39.209	90.948	0.99855	0.00041958	1.0028	0.14531	0.10095	5.2269e-05	0.0020521	0.0093765	6.9743	4.1313e-05	0.1625	0.10377
'fake'	5.5646e-05	40.288	90.676	0.99897	0.0004452	1.002	0.14498	0.10007	5.5646e-05	0.0019792	0.009064	6.917	4.993e-05	0.16027	0.008547
'fake'	6.8375e-05	38.421	89.782	0.99843	0.00042914	1.003	0.14545	0.11204	0.0042028	0.0016927	0.0080838	10.245	5.6433e-05	0.16859	0.10638

Table 17 : Suject-4 LDA results

LDA:

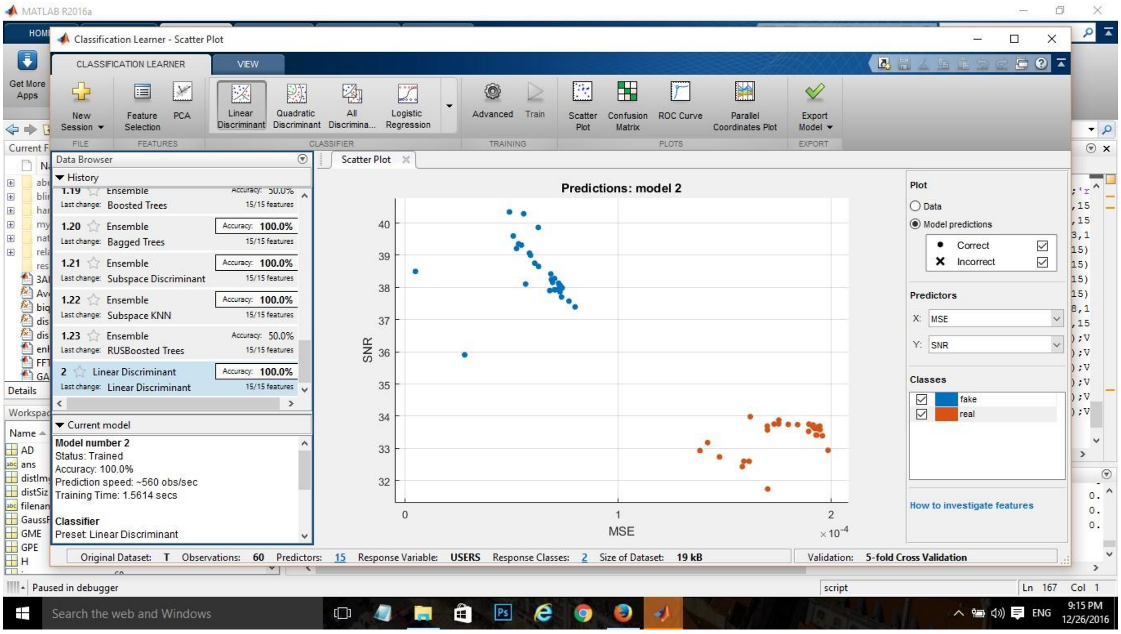


Figure C.1: Scatter plot figure

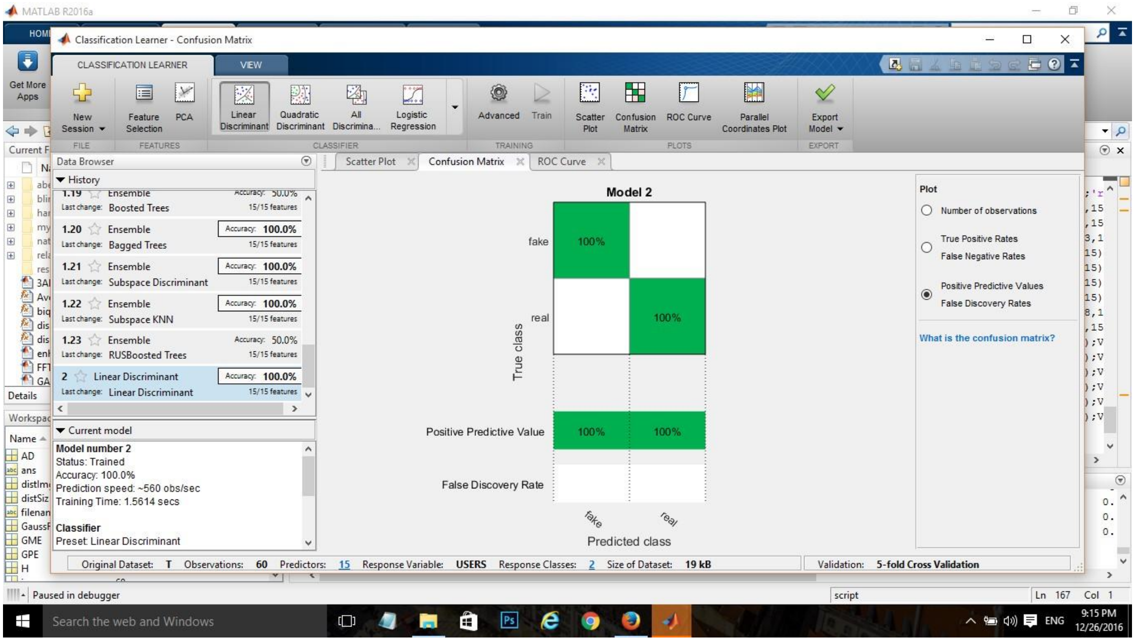


Figure C.2: Confusion matrix figure

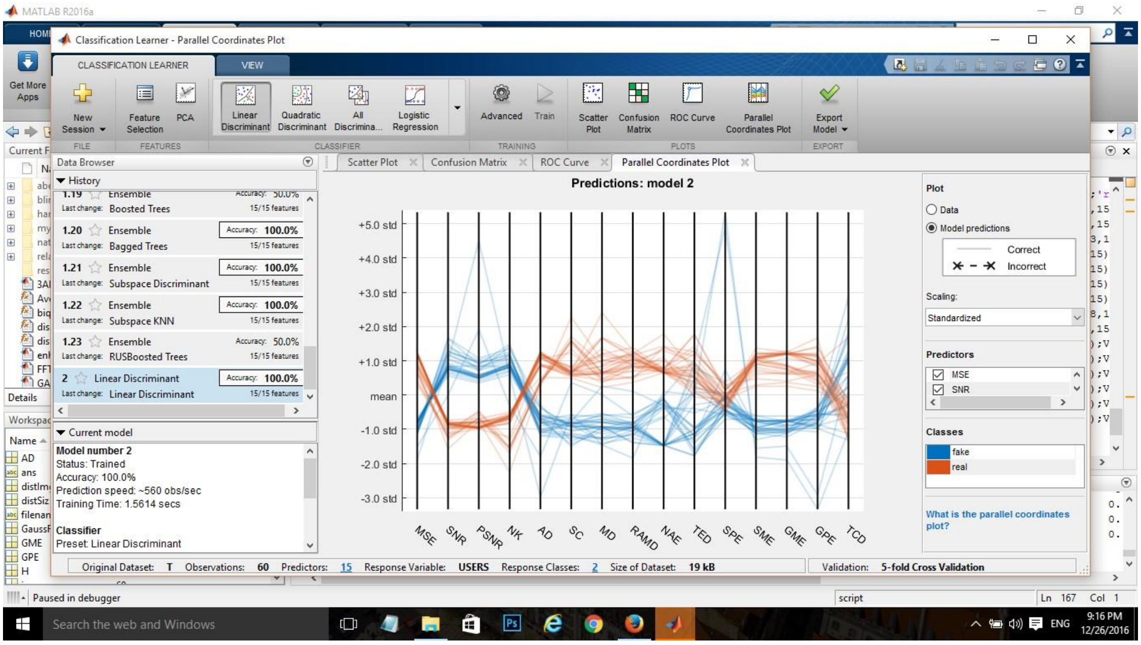


Figure C.3: Parallel coordinate plot

Suject-4 LDA results plotted on Table: C.1, scatter plot Figure: C.1, confusion matrix Figure C.2, parallel coordinate plot Figure C.3

QDA:

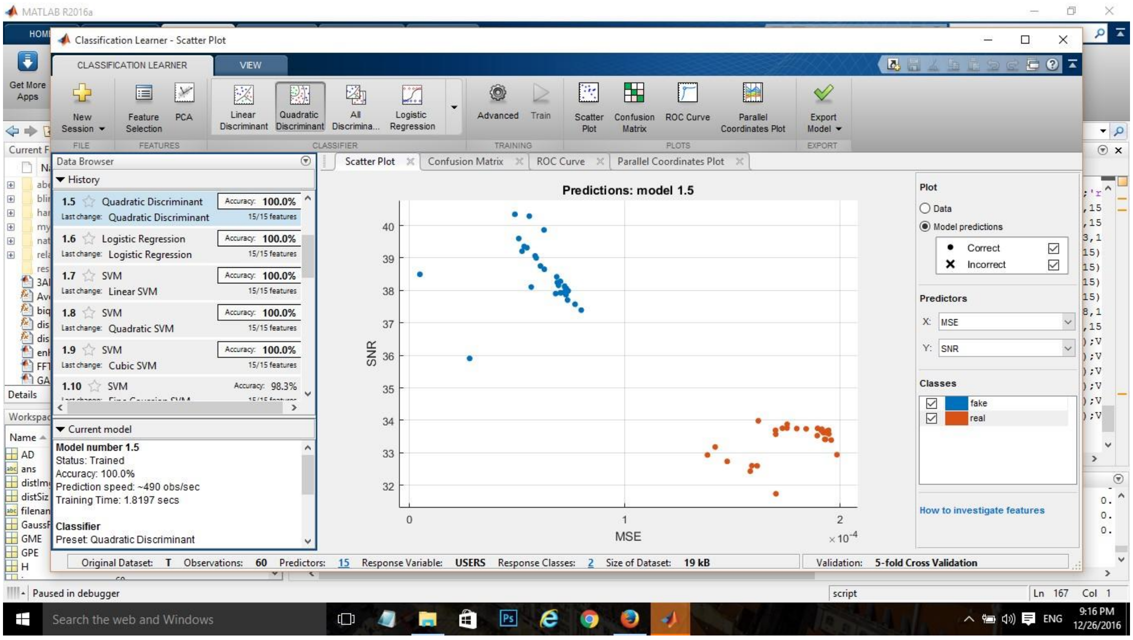


Figure: C.4: Scatter plot

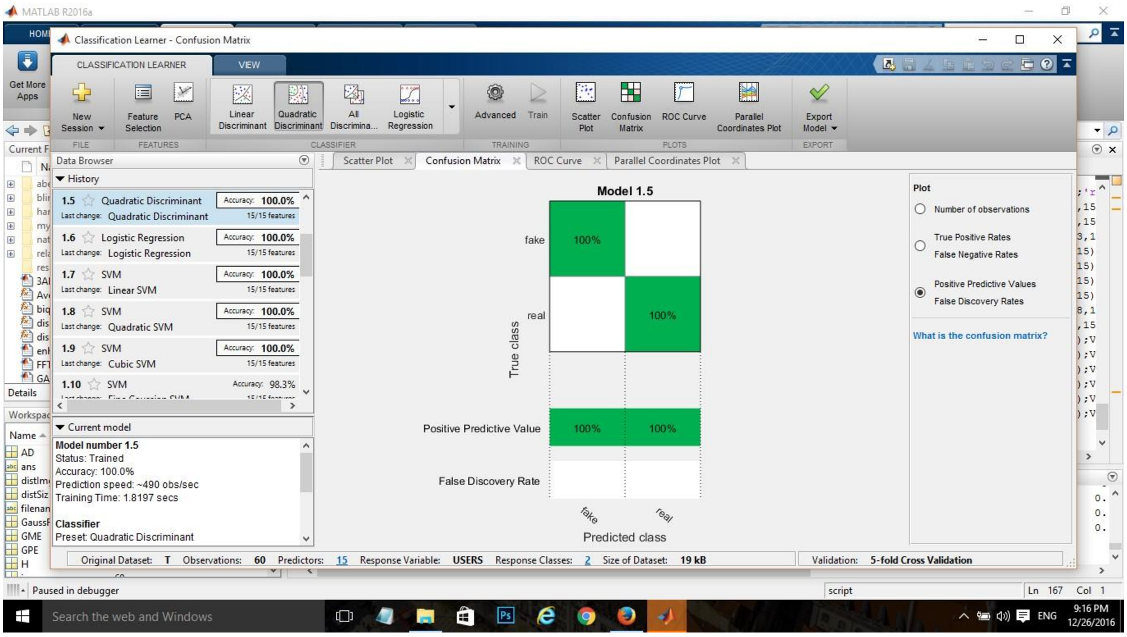


Figure C.5: Confusion matrix

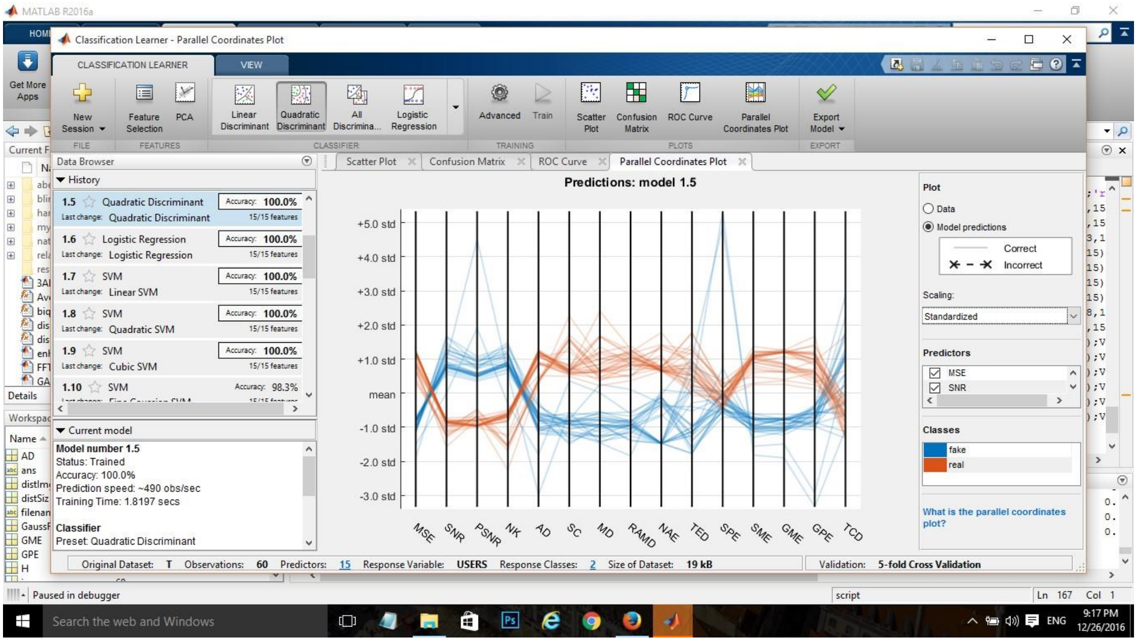


Figure C.6: Parallel coordinate plot

Subject-4 QDA results plotted on scatter plot Figure C.4, confusion matrix Figure C.5, parallel coordinate plot Figure C.6

LINEAR SVM:

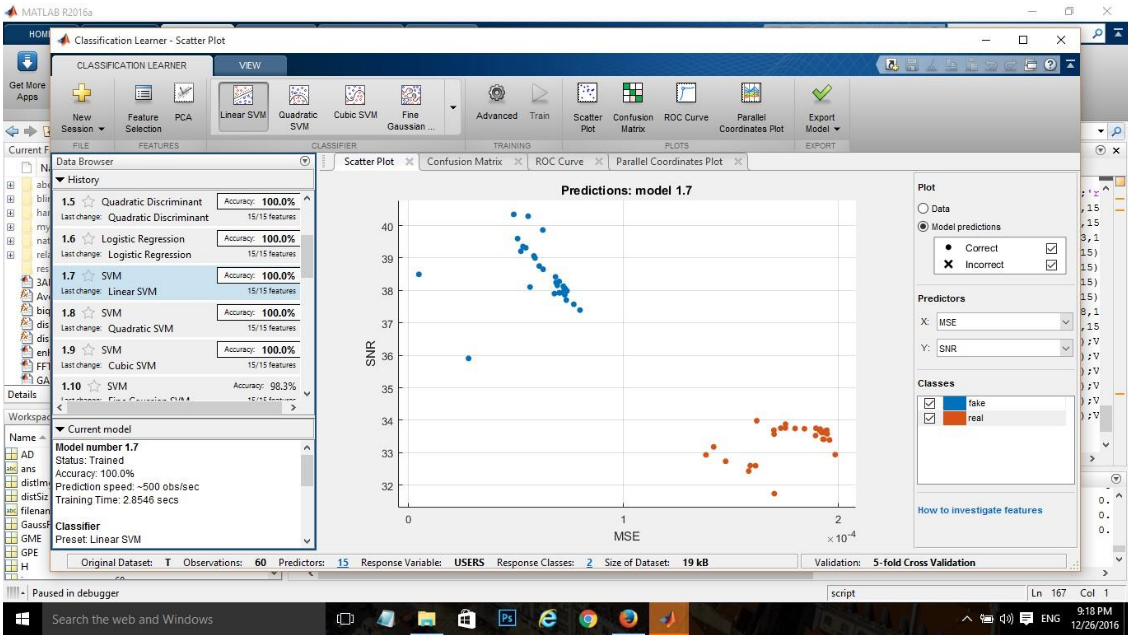


Figure C.7: Scatter plot

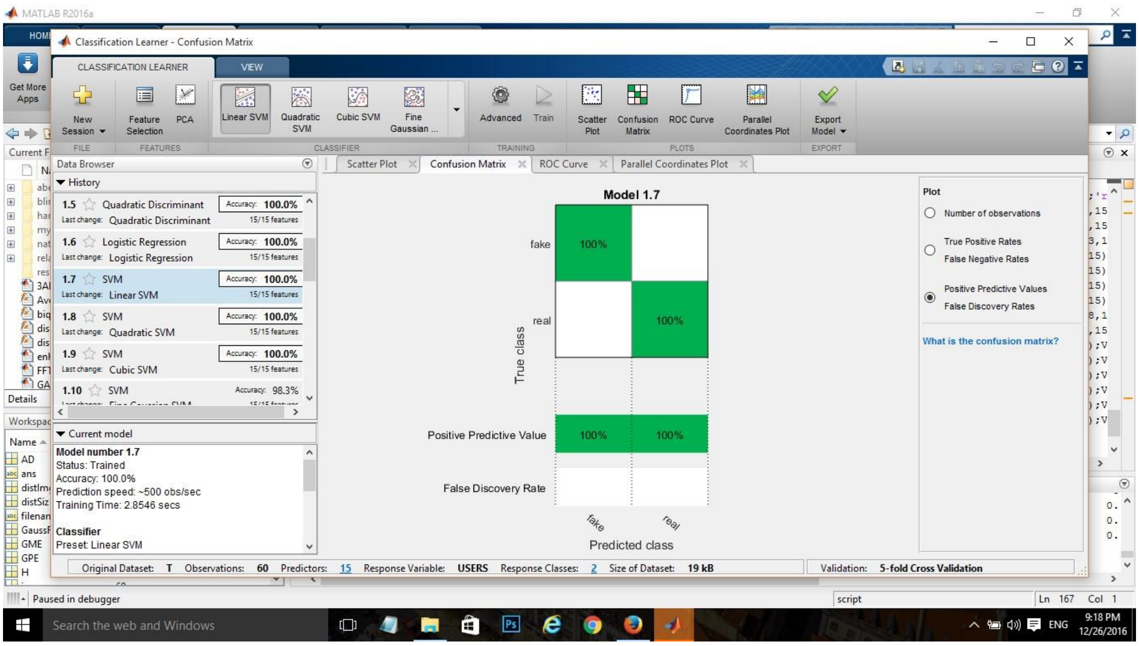


Figure C.8: Confusion matrix

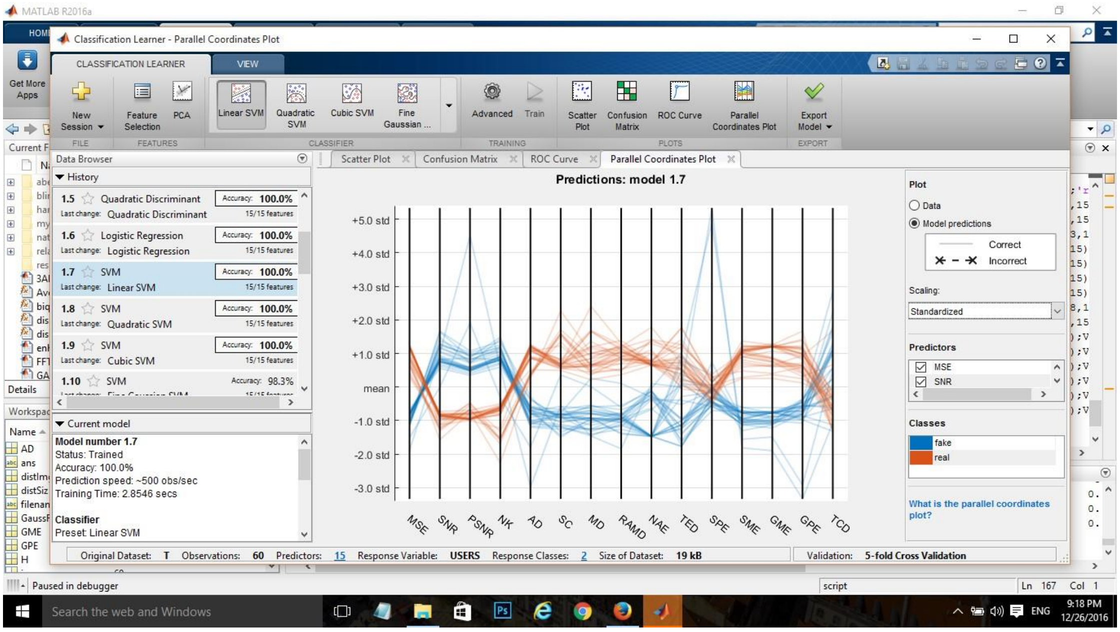


Figure C.9: Parallel coordinate

Dataset-4 linear SVM results plotted on scatter plot Figure C.7, confusion matrix Figure C.8, parallel coordinate Figure C.9

QUADRATIC SVM:

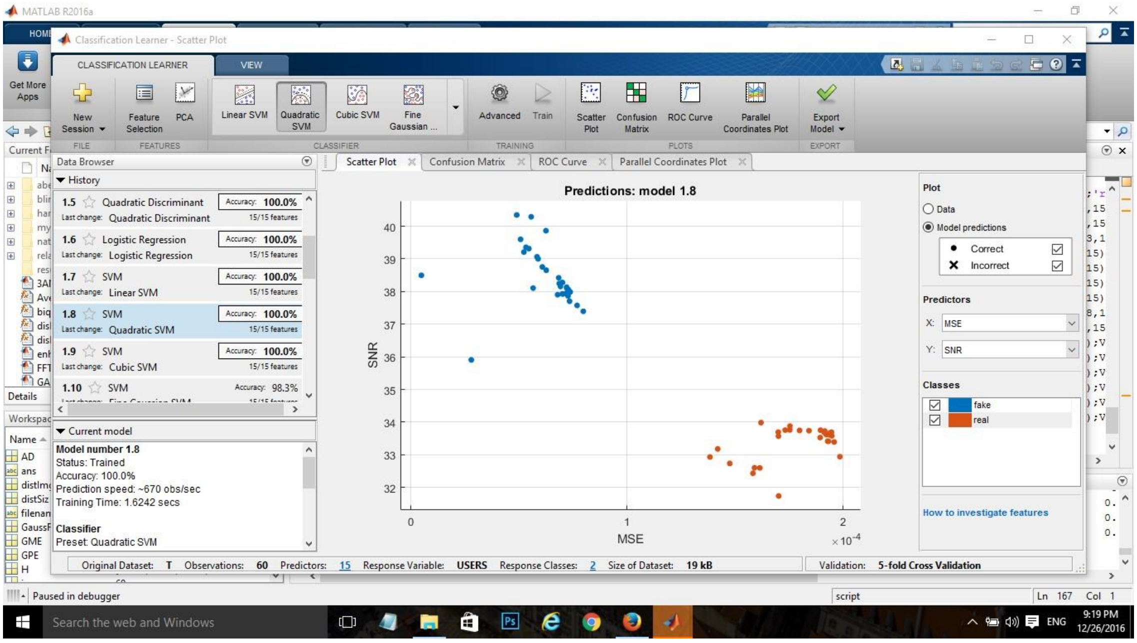


Figure C.10: Scatter plot

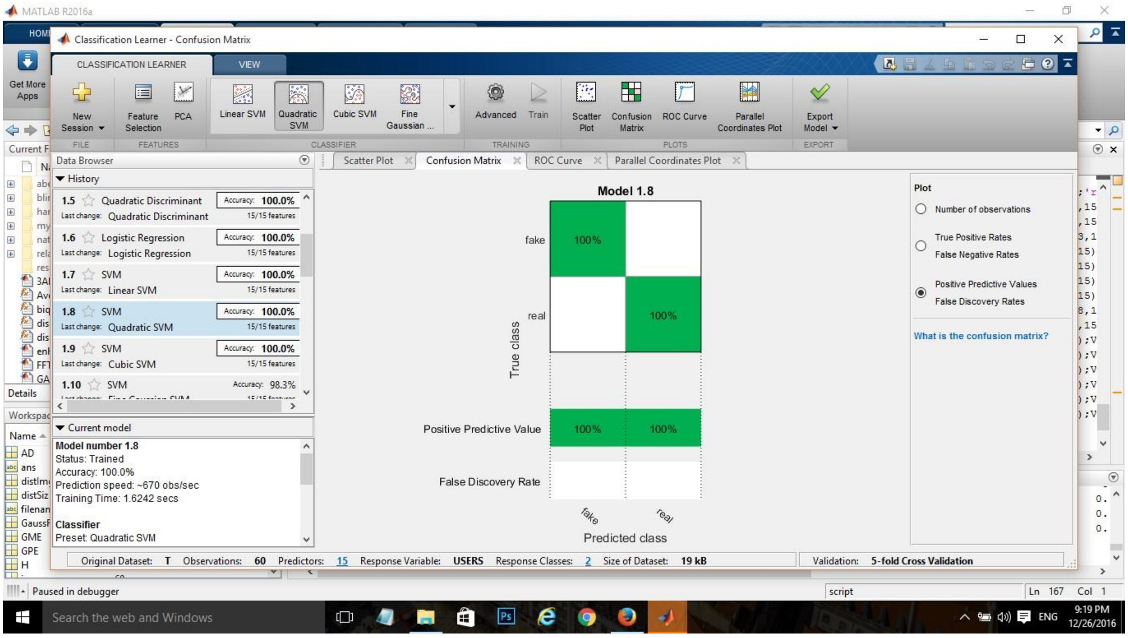


Figure C.11: Confusion matrix

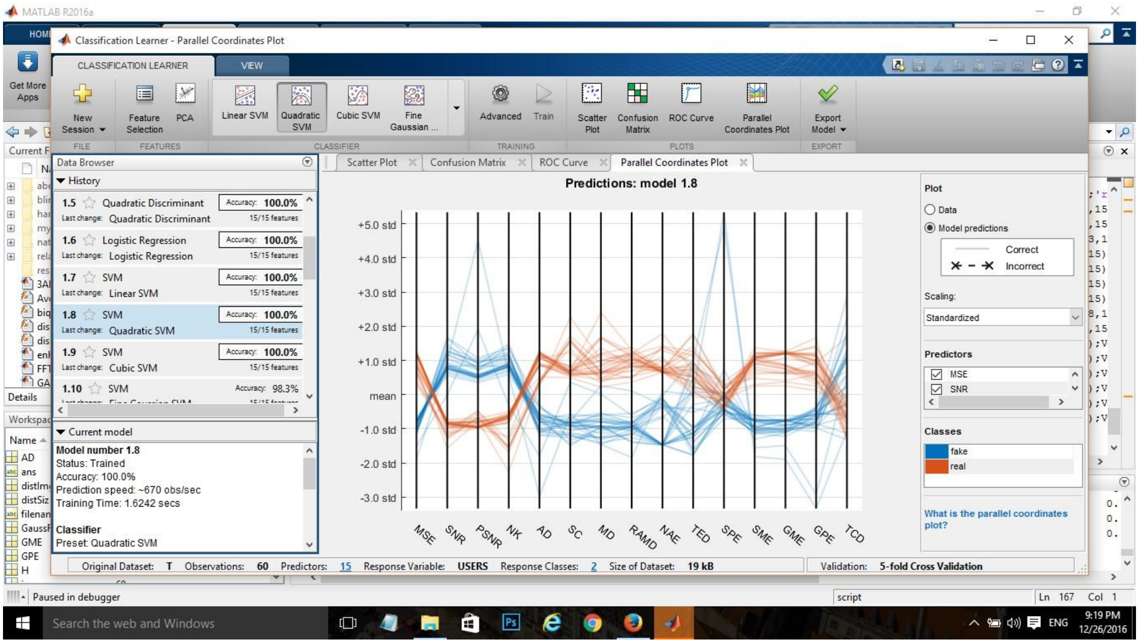


Figure C.12: Parallel coordinate plot

Dataset-4 quadratic SVM results plotted on scatter plot Figure: C.10, confusion matrix Figure C.11, parallel coordinate plot Figure C.12

LOGISTIC REGRESSION:

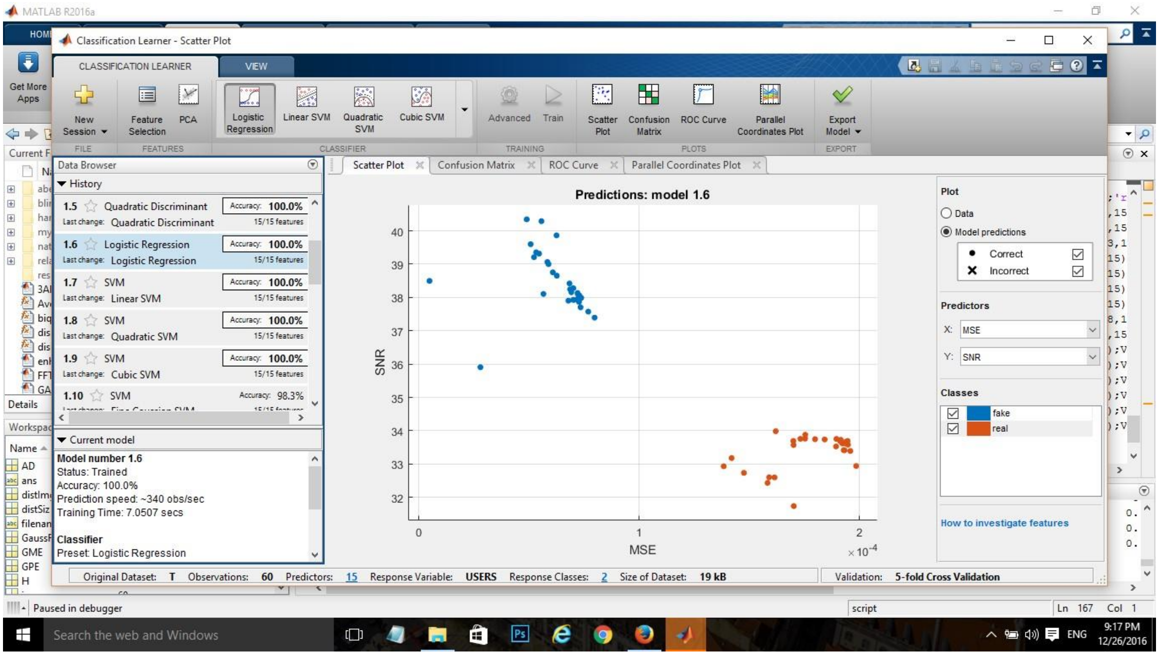


Figure C.13: Scatter plot

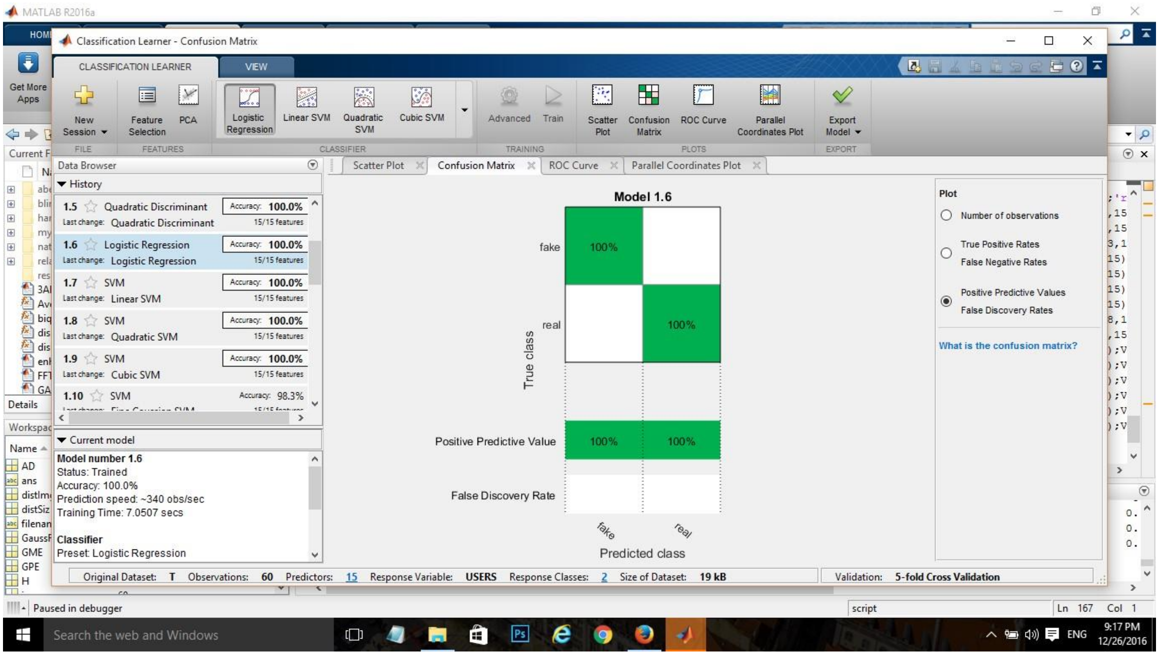


Figure C.14: Confusion matrix

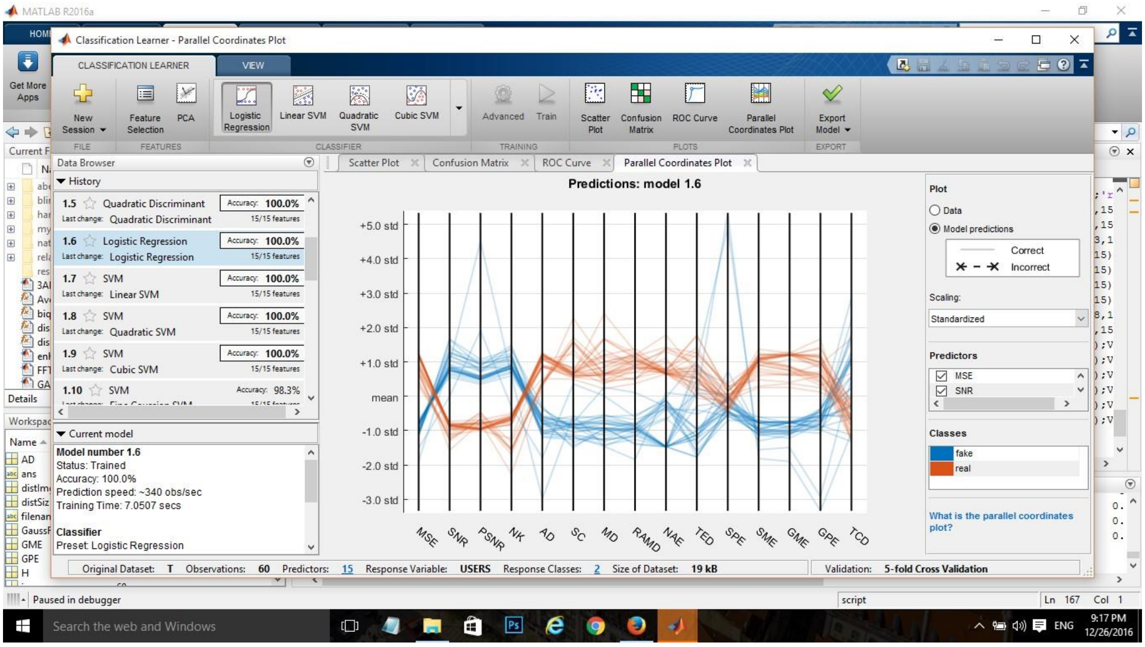


Figure C.15: Parallel coordinate plot

Dataset-4 logistic regression results plotted on scatter plot Figure C.13, confusion matrix Figure C.14, parallel coordinate plot Figure C.15

Appendix C2

BEST-5

This table shows the result of best-5 image quality measurements calculations with 30 real face images and 30 fake samples, for training.

60 face image samples using Best-5 (IQA) measures results

USERS	SNR	PSNR	GME	RAMD	NAE
'real'	34.559	88.877	9.3597e-05	0.12007	0.00593
'real'	34.555	89.308	8.3676e-05	0.11753	0.0060485
'real'	32.776	89.532	7.6425e-05	0.13075	0.0083913
'real'	34.53	88.822	9.3662e-05	0.12344	0.0057917
'real'	34.884	88.795	9.5253e-05	0.12169	0.0053178
'real'	35.024	88.807	9.4725e-05	0.12402	0.0051799
'real'	35.361	88.696	9.7598e-05	0.11991	0.0047879
'real'	35.205	88.786	9.6072e-05	0.12807	0.0049823
'real'	34.886	88.753	9.6908e-05	0.11976	0.0052794
'real'	35.076	88.763	9.5803e-05	0.12522	0.0051271
'real'	34.203	88.698	9.6514e-05	0.135	0.0062645
'real'	34.421	88.666	9.9036e-05	0.12585	0.0058685
'real'	34.895	88.66	9.9127e-05	0.12048	0.0052329
'real'	35.129	88.587	9.9566e-05	0.13035	0.0051578
'real'	34.883	88.624	9.7987e-05	0.1199	0.0053251
'real'	34.229	88.312	0.00010356	0.13645	0.0061778
'real'	33.572	85.825	0.00015093	0.19082	0.0081743
'real'	33.878	85.69	0.0001549	0.18654	0.0077468
'real'	33.745	85.581	0.00015767	0.16788	0.0079178
'real'	33.838	85.663	0.00015458	0.17485	0.0077944
'real'	33.617	85.291	0.00017472	0.18138	0.0077732
'real'	33.756	85.745	0.00015593	0.17936	0.0080104
'real'	33.762	85.692	0.00015638	0.17326	0.0079211
'real'	33.695	85.826	0.0001493	0.17841	0.0081397
'real'	33.681	85.313	0.00017538	0.17749	0.0076312
'real'	33.416	85.27	0.00017647	0.19771	0.0080081
'real'	33.393	85.211	0.00018014	0.19304	0.0079388
'real'	33.638	85.272	0.00017547	0.18345	0.0076866
'real'	32.435	86.137	0.00014209	0.18983	0.010396
'real'	33.179	86.608	0.00012931	0.16744	0.0088657
'fake'	35.493	88.215	0.000106	0.14331	0.0051696
'fake'	36.243	89.046	8.3146e-05	0.155	0.0045605
'fake'	35.345	89.022	8.2103e-05	0.14201	0.0056047
'fake'	35.937	88.658	8.7569e-05	0.16182	0.0053674
'fake'	34.967	88.921	8.4007e-05	0.15956	0.0058607
'fake'	35.253	87.763	0.00010958	0.15927	0.006157
'fake'	36.952	88.742	9.3703e-05	0.1873	0.0040661
'fake'	35.597	88.149	9.9857e-05	0.1699	0.0052483
'fake'	35.421	88.561	9.4674e-05	0.13973	0.0055972
'fake'	37.175	88.559	9.457e-05	0.15703	0.0039232
'fake'	38.497	101.31	3.9738e-06	0.084648	0.0056693
'fake'	36.283	90.179	5.4458e-05	0.15503	0.00542
'fake'	36.63	90.787	5.0339e-05	0.1522	0.0050557
'fake'	37.208	89.719	7.1426e-05	0.17229	0.0039925
'fake'	35.276	87.736	0.00012021	0.2081	0.00010953
'fake'	36.426	87.733	0.00011548	0.1646	0.0001096
'fake'	35.907	93.669	2.641e-05	0.12786	2.7936e-05
'fake'	38.105	90.607	5.0944e-05	0.12984	5.6548e-05
'fake'	37.902	89.814	6.6474e-05	0.15286	6.7866e-05

'fake'	37.704	89.474	6.8378e-05	0.1192	7.3403e-05
'fake'	37.393	89.114	7.4845e-05	0.12049	7.9749e-05
'fake'	37.575	89.272	6.581e-05	0.12016	7.6885e-05
'fake'	40.346	91.234	4.6254e-05	0.11298	4.8946e-05
'fake'	39.865	90.173	5.5334e-05	0.11609	6.2492e-05
'fake'	37.927	89.65	6.0384e-05	0.11537	7.0479e-05
'fake'	37.949	89.557	6.5327e-05	0.11883	7.2001e-05
'fake'	38.243	89.764	6.548e-05	0.12542	6.8657e-05
'fake'	38.102	89.549	6.6047e-05	0.11568	7.2132e-05
'fake'	39.001	90.431	5.1857e-05	0.10772	5.8876e-05
'fake'	39.356	90.868	4.8106e-05	0.10517	0.0038863

Table 18 : Best-5 LDA results

LDA:

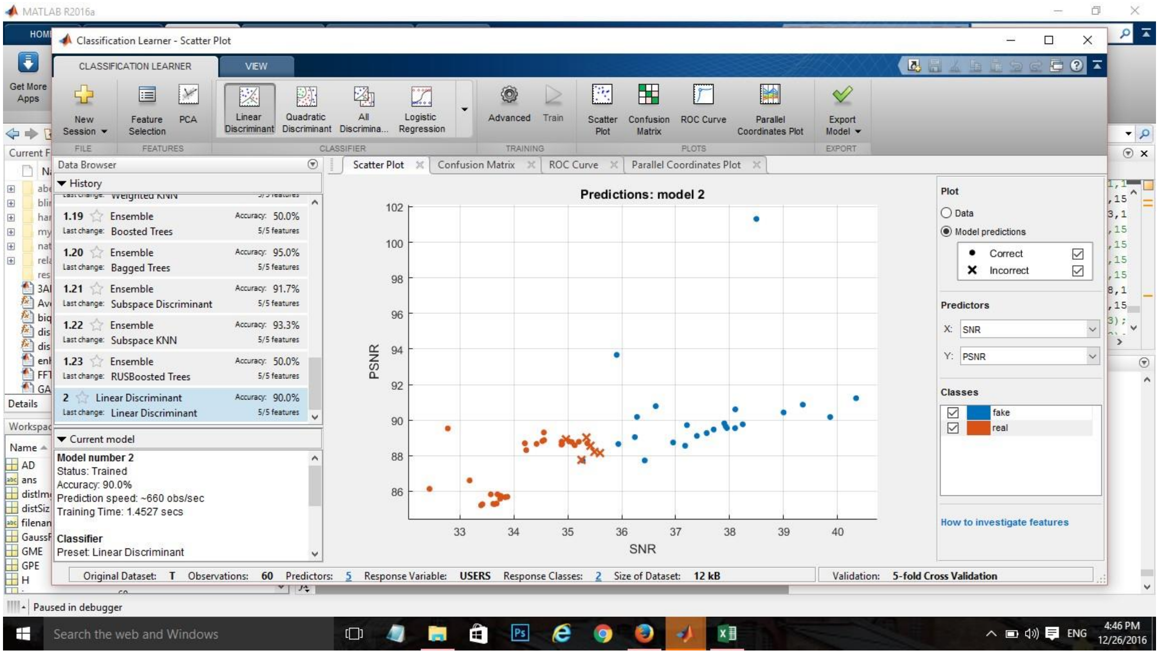


Figure C.16: Scatter plot

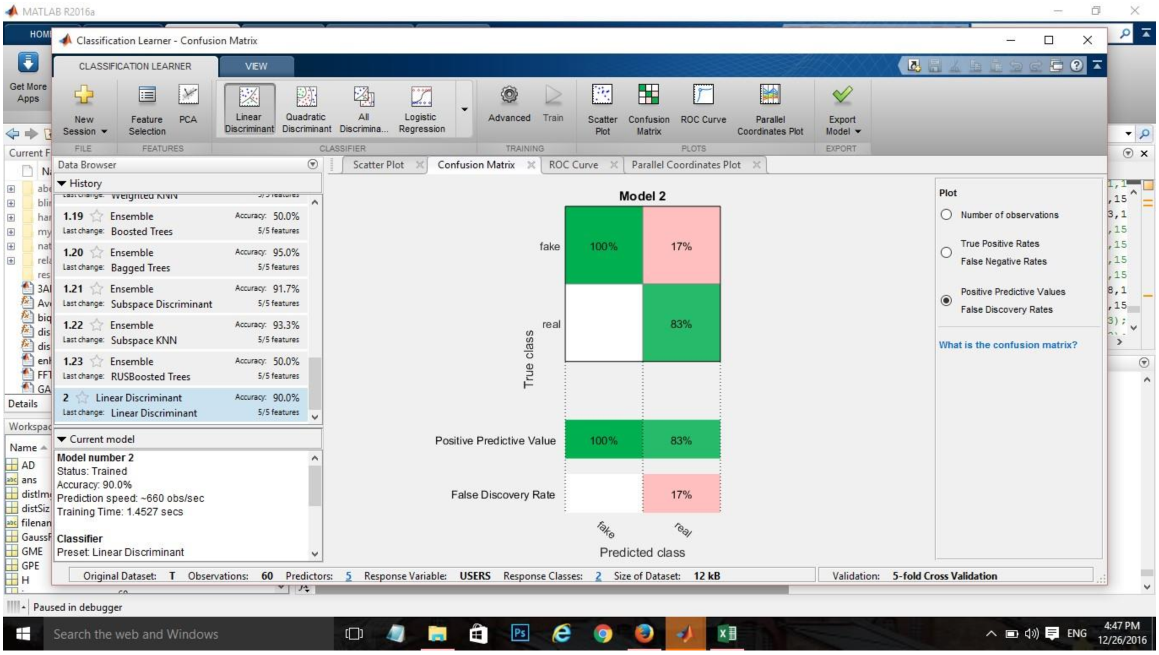


Figure C.17: Confusion matrix

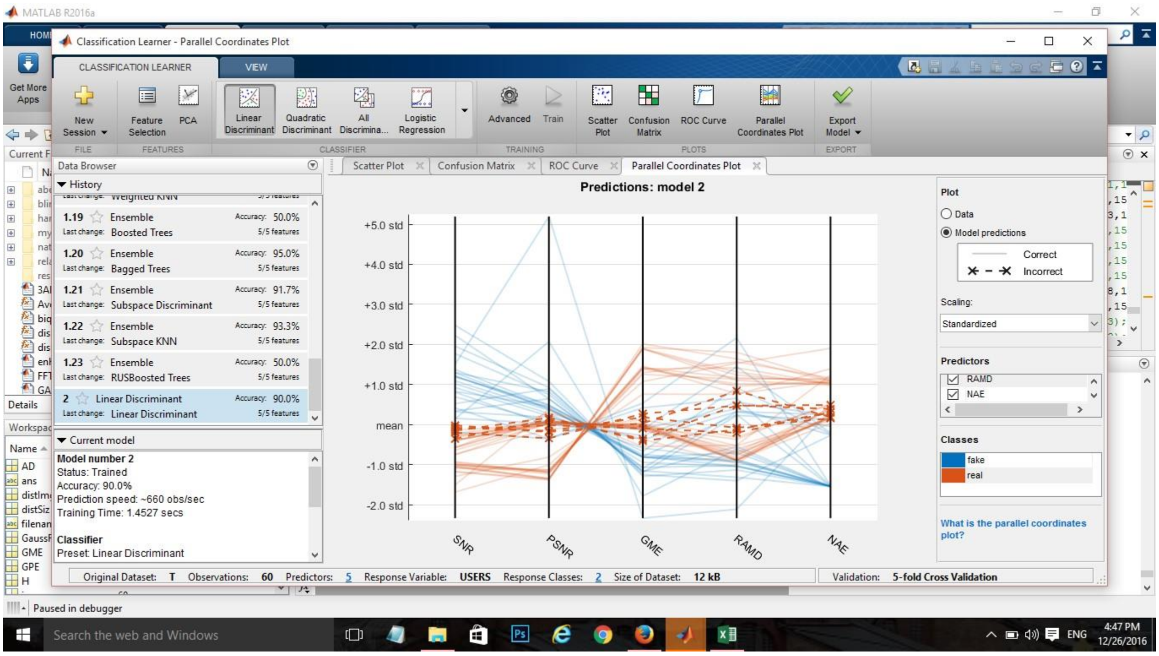


Figure C.18: Parallel coordinate

Best-5 LDA results plotted on scatter plot Figure C.16, confusion matrix Figure C.17, parallel coordinate Figure C.18

QDA:

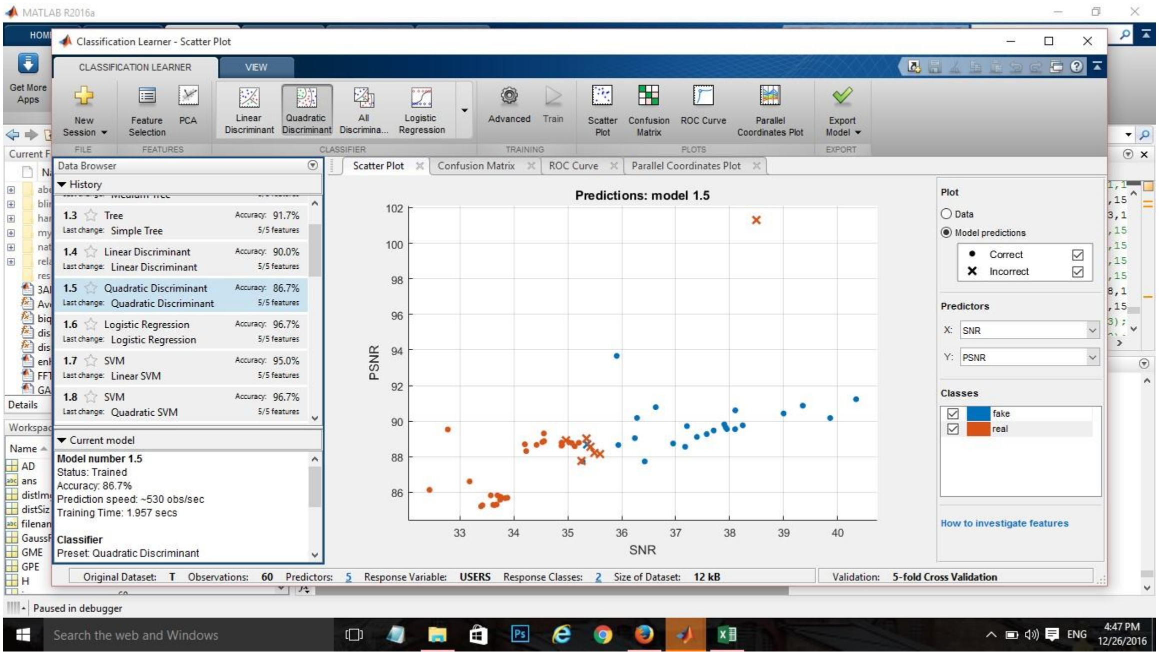


Figure C.19: Scatter plot

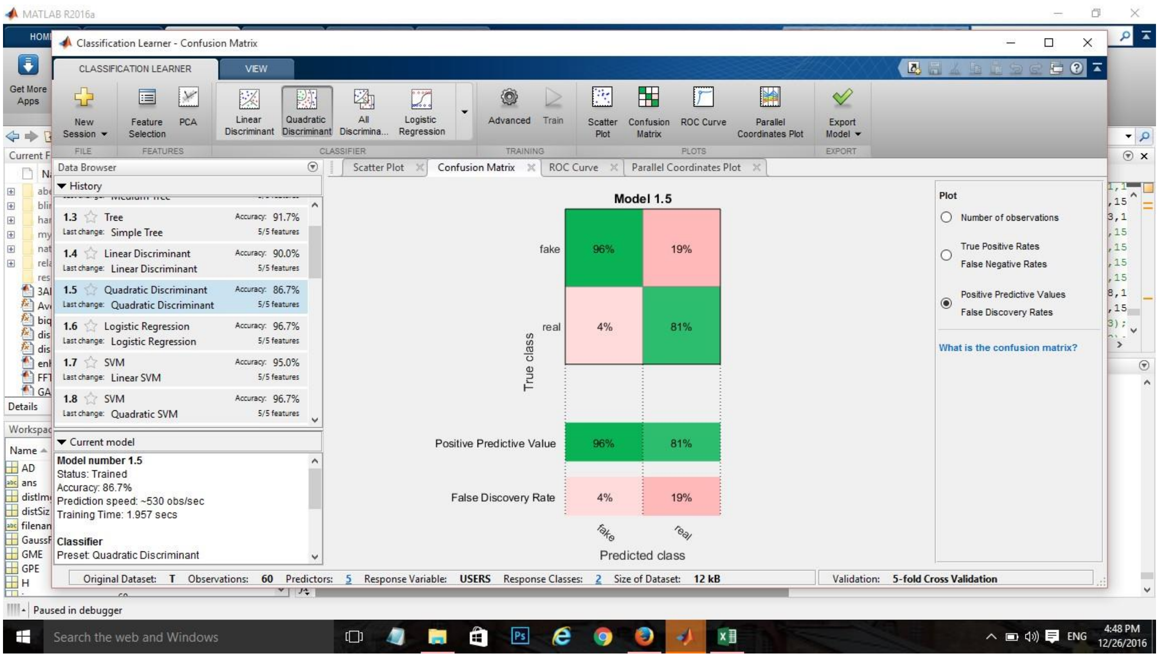


Figure C.20: Confusion matrix

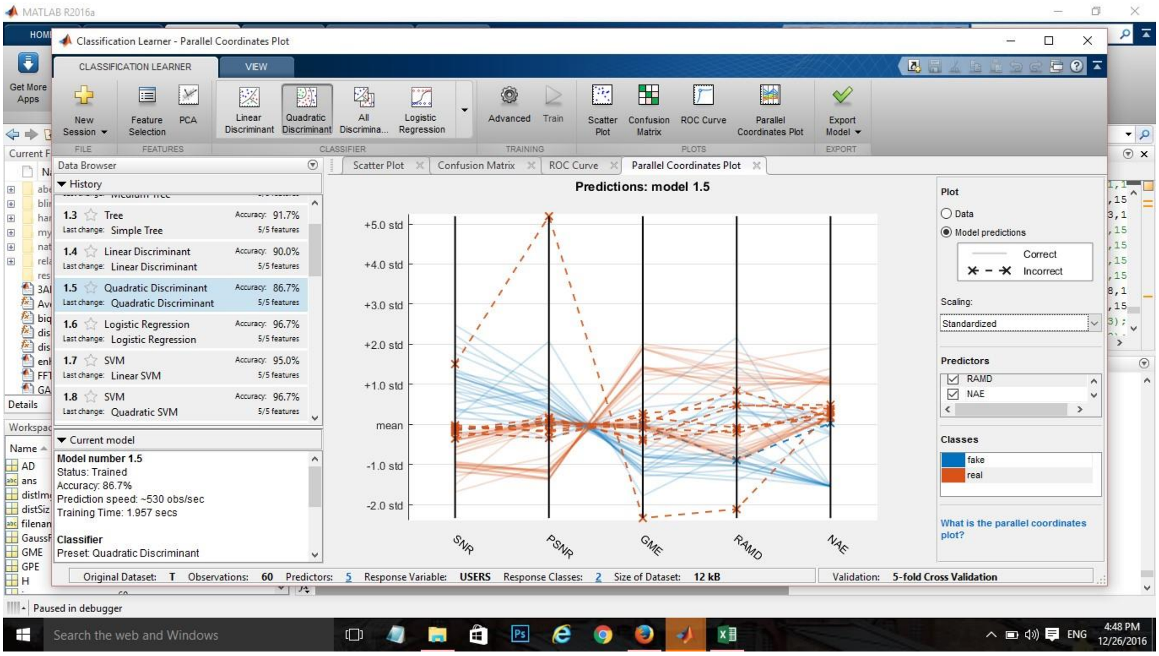


Figure C.21: Parallel coordinate

Best-5 QDA results plotted on scatter plot Figure C.19, confusion matrix, parallel coordinate

LINEAR SVM:

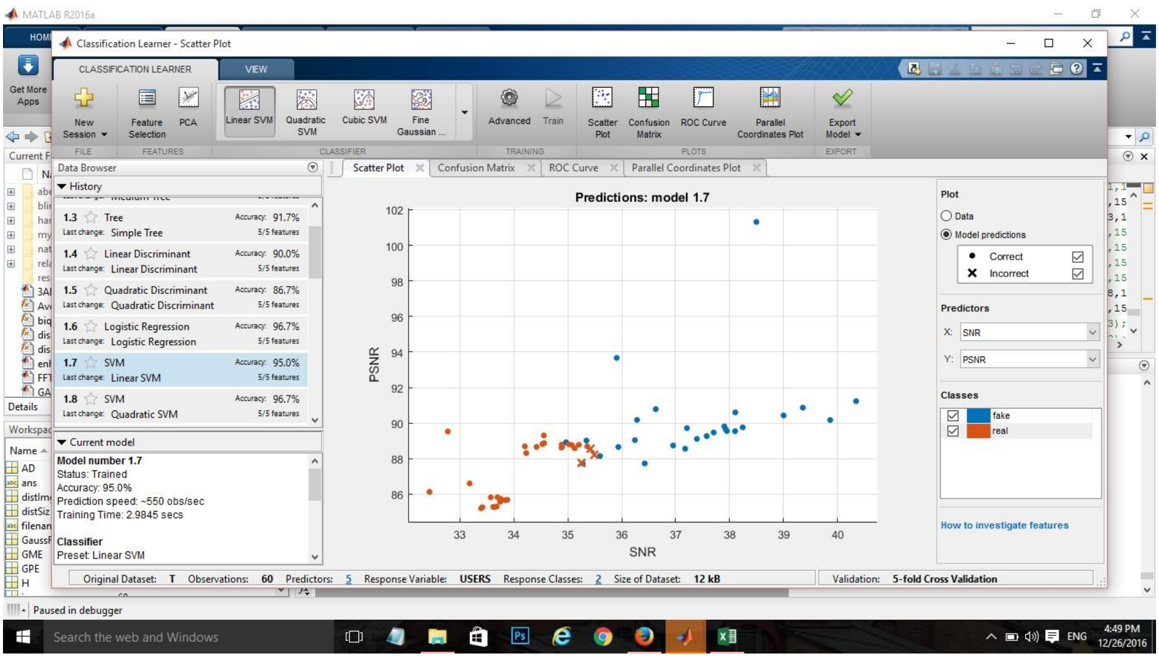


Figure C.22: Scatter plot

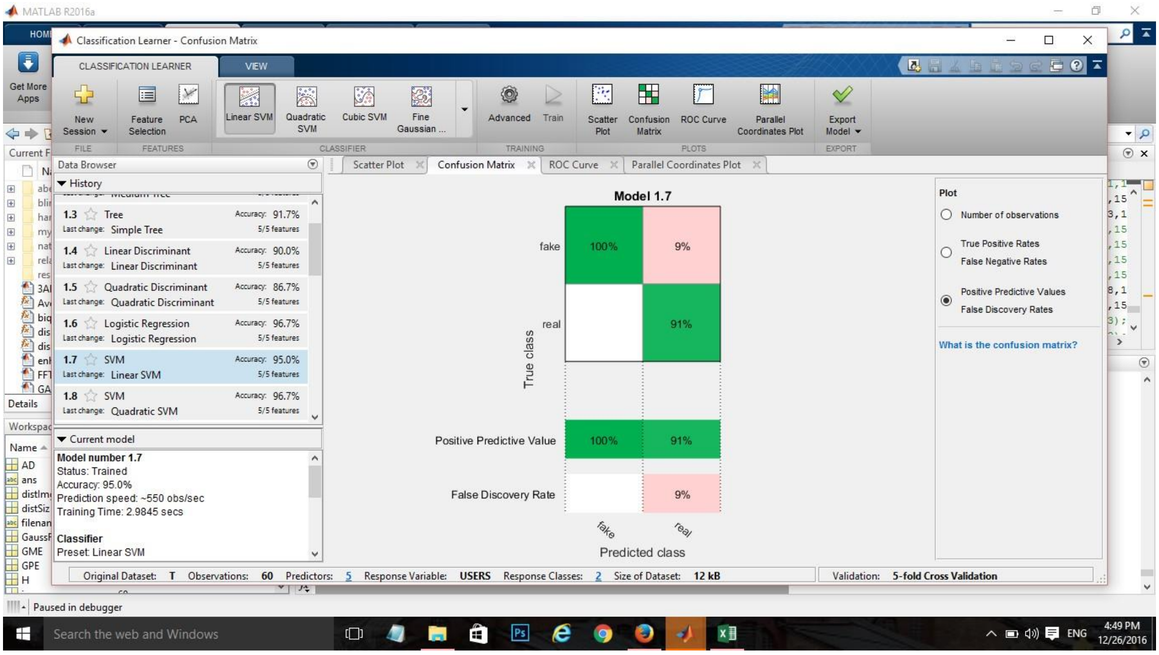


Figure C.23: Confusion matrix

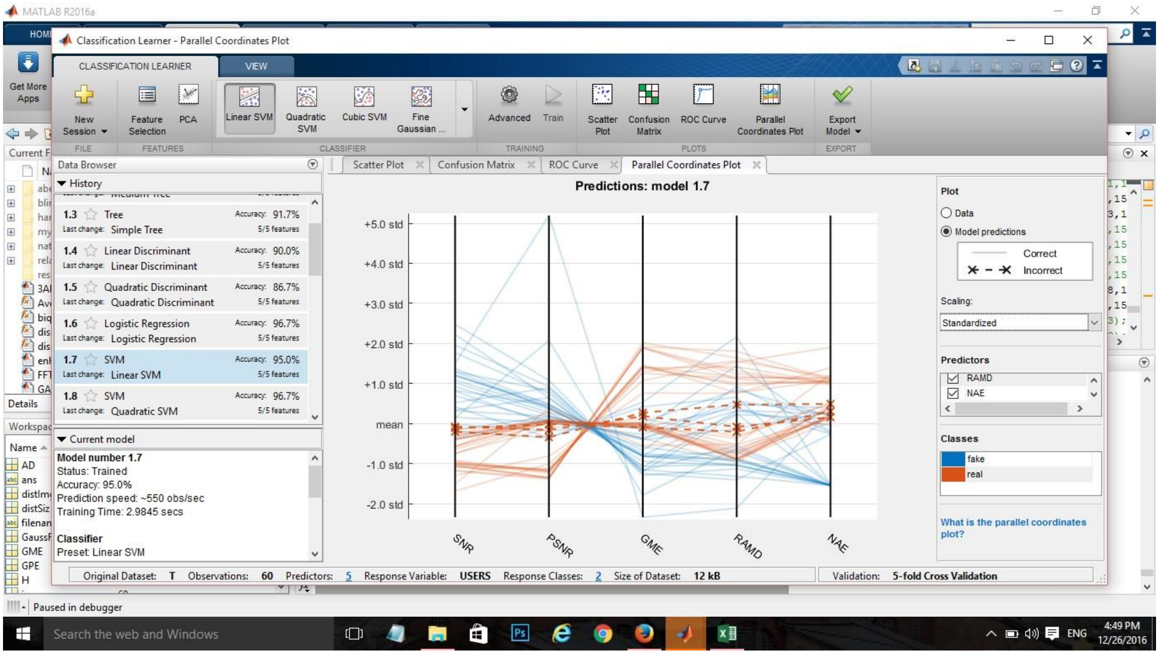


Figure C.23: Parallel coordinate

Best-5 Linear SVM results plotted on scatter plot Figure C.21, confusion matrix

Figure C.22, parallel coordinate Figure C.23

QUADRATIC SVM:

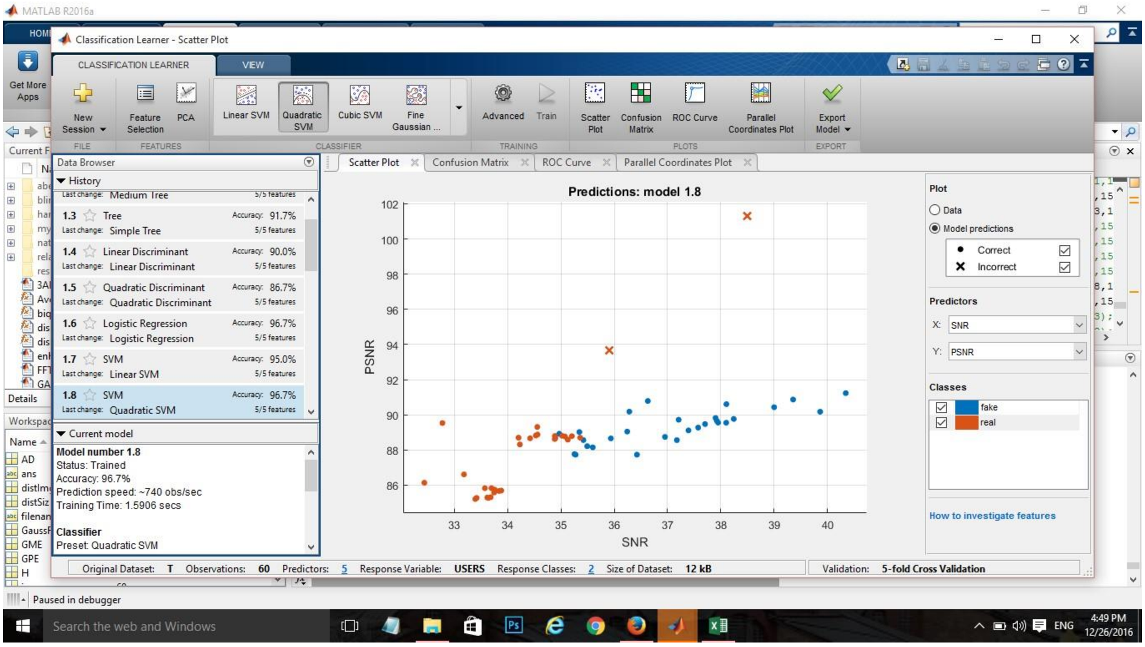


Figure C.24: Scatter plot

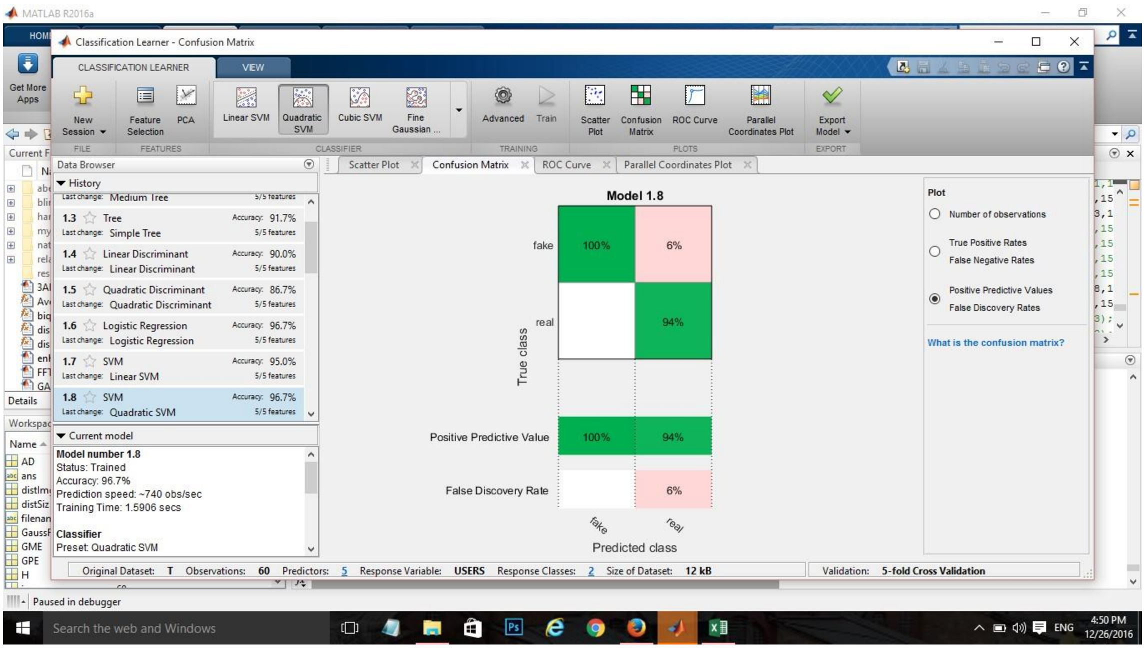


Figure C.25: Confusion matrix

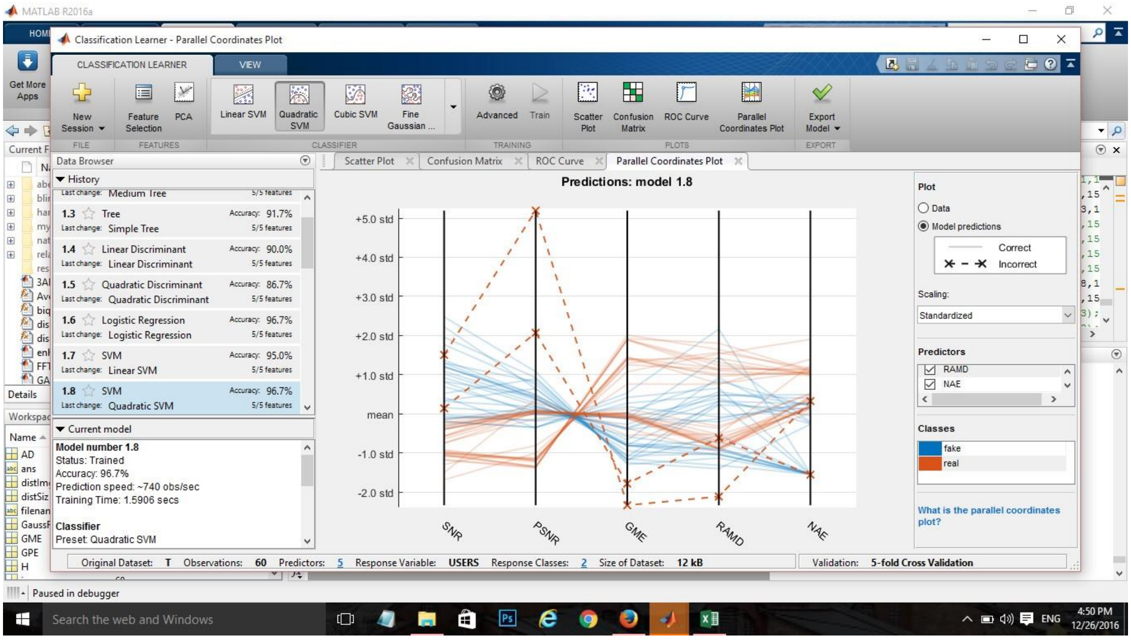


Figure C.26: Parallel coordinate

Best-5 Quadratic SVM results plotted on Figure C.24 scatter plot, confusion matrix Figure C.25, parallel coordinate Figure C.26

LOGISTIC REGRESSION:

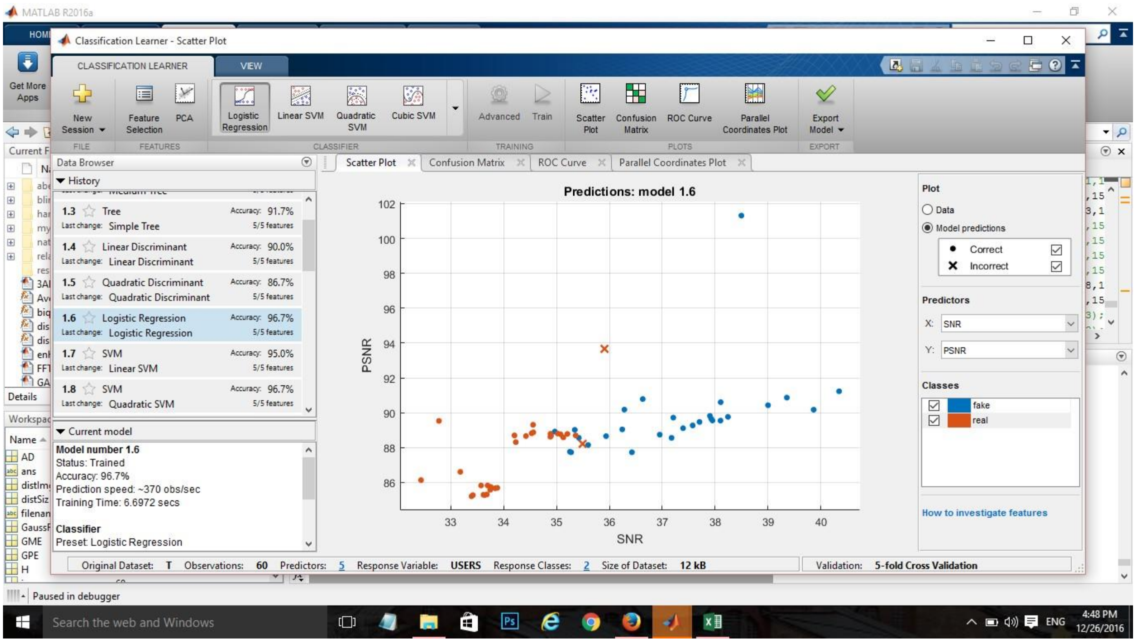


Figure C.27: Scatter plot

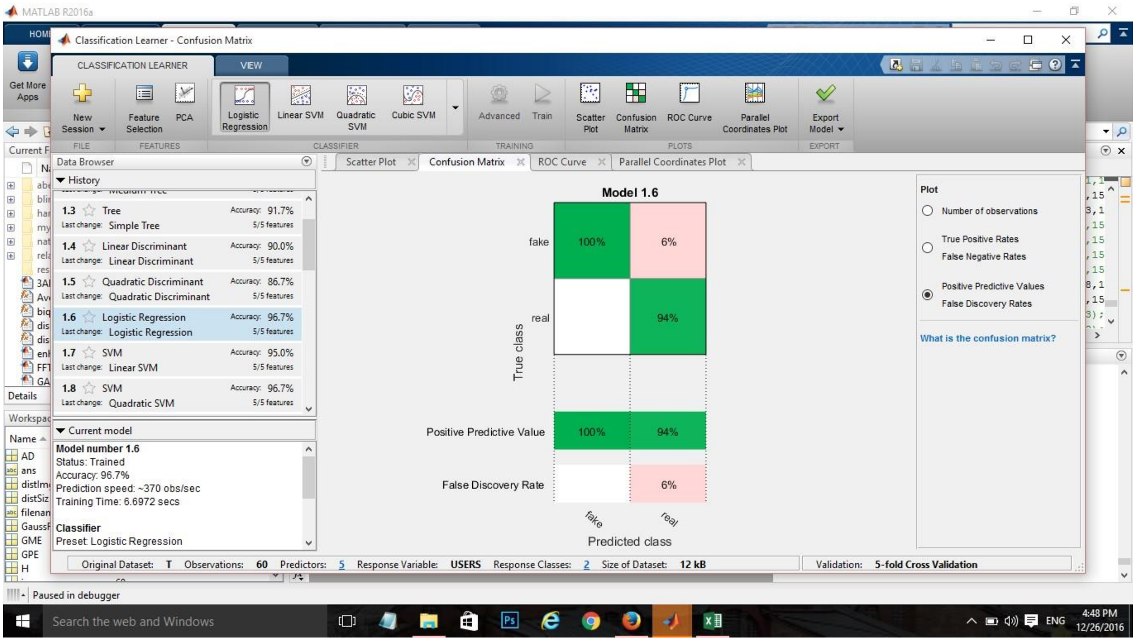


Figure C.28: Confusion matrix

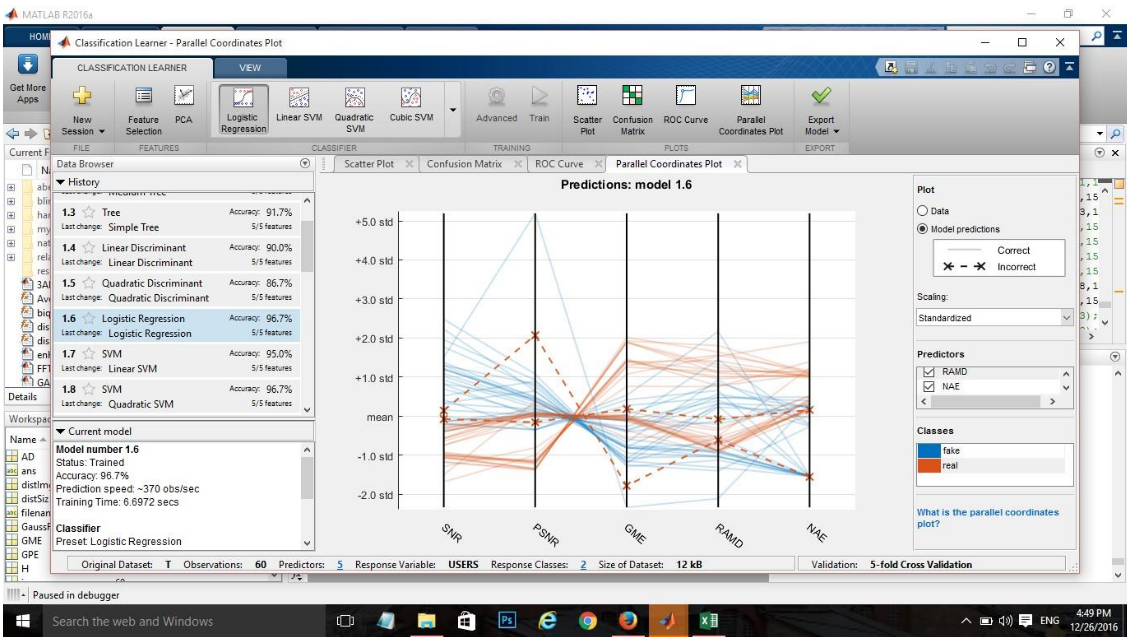


Figure C.29: Parallel coordinate

Best-5 logistic regression results plotted on scatter plot Figure C.27, confusion matrix Figure C.28, parallel coordinate Figure C.29

Appendix C3

BEST-10:

This table shows the result of best-5 image quality measurements calculations with 30 real face images and 30 fake samples. 60 face image samples using Best-10 (IQA) measures results

USERS	MSE	SNR	PSNR	AD	SC	MD	RAMD	NAE	SPE	GME
'real'	8.4209e-05	34.559	88.877	0.00064206	1.0063	0.20167	0.12007	0.00593	0.018559	9.3597e-05
'real'	7.625e-05	34.555	89.308	0.00061616	1.0064	0.20167	0.11753	0.0060485	0.017381	8.3676e-05
'real'	7.243e-05	32.776	89.532	0.00058189	1.0094	0.20167	0.13075	0.0083913	0.022029	7.6425e-05
'real'	8.5278e-05	34.53	88.822	0.00063461	1.0063	0.20167	0.12344	0.0057917	0.021296	9.3662e-05
'real'	8.5812e-05	34.884	88.795	0.00064549	1.0058	0.20167	0.12169	0.0053178	0.018361	9.5253e-05
'real'	8.5573e-05	35.024	88.807	0.00064688	1.0056	0.20167	0.12402	0.0051799	0.01955	9.4725e-05
'real'	8.779e-05	35.361	88.696	0.00066055	1.0052	0.20167	0.11991	0.0047879	0.016268	9.7598e-05
'real'	8.6004e-05	35.205	88.786	0.00066703	1.0055	0.20167	0.12807	0.0049823	0.018106	9.6072e-05
'real'	8.6658e-05	34.886	88.753	0.00066096	1.0058	0.20167	0.11976	0.0052794	0.016649	9.6908e-05
'real'	8.6452e-05	35.076	88.763	0.00066368	1.0056	0.20167	0.12522	0.0051271	0.018175	9.5803e-05
'real'	8.7757e-05	34.203	88.698	0.00062173	1.0069	0.20167	0.135	0.0062645	0.021273	9.6514e-05
'real'	8.8399e-05	34.421	88.666	0.00067401	1.0065	0.20167	0.12585	0.0058685	0.018366	9.9036e-05
'real'	8.8535e-05	34.895	88.66	0.00066655	1.0059	0.20167	0.12048	0.0052329	0.023758	9.9127e-05
'real'	9.0019e-05	35.129	88.587	0.00065226	1.0056	0.20167	0.13035	0.0051578	0.017134	9.9566e-05
'real'	8.9271e-05	34.883	88.624	0.0006397	1.0058	0.20167	0.1199	0.0053251	0.018866	9.7987e-05
'real'	9.5909e-05	34.229	88.312	0.00063929	1.0066	0.20167	0.13645	0.0061778	0.019357	0.00010356
'real'	9.4493e-05	34.597	88.377	0.00064353	1.0061	0.20167	0.13461	0.0054946	0.020816	0.00010296
'real'	0.00013839	32.934	86.72	0.00061522	1.0075	0.19492	0.17396	0.0092871	0.012271	0.00012298
'real'	0.00014198	33.179	86.608	0.00062833	1.0071	0.19256	0.16744	0.0088657	0.01416	0.00012931
'real'	0.00016147	32.598	86.05	0.00061549	1.0074	0.20097	0.18342	0.0099305	0.013656	0.00014512
'real'	0.00017006	33.572	85.825	0.00070483	1.006	0.21368	0.19082	0.0081743	0.015239	0.00015093
'real'	0.0001754	33.878	85.69	0.00070713	1.0056	0.20167	0.18654	0.0077468	0.017633	0.0001549
'real'	0.00017986	33.745	85.581	0.00070018	1.0056	0.19386	0.16788	0.0079178	0.015126	0.00015767
'real'	0.0001765	33.838	85.663	0.00068959	1.0056	0.19081	0.17485	0.0077944	0.01914	0.00015458
'real'	0.00019228	33.617	85.291	0.00076428	1.0059	0.21396	0.18138	0.0077732	0.0098173	0.00017472
'real'	0.0001732	33.756	85.745	0.00071376	1.0058	0.22671	0.17936	0.0080104	0.0152	0.00015593
'real'	0.00017533	33.762	85.692	0.00070509	1.0057	0.20167	0.17326	0.0079211	0.015011	0.00015638
'real'	0.00017001	33.695	85.826	0.00070015	1.0059	0.20167	0.17841	0.0081397	0.016233	0.0001493
'real'	0.00019131	33.681	85.313	0.00075223	1.0058	0.19907	0.17749	0.0076312	0.0098657	0.00017538
'real'	0.00019291	33.415	85.277	0.00075341	1.0061	0.1908	0.17782	0.008108	0.0095362	0.00017559
'fake'	4.8054e-06	38.497	101.31	5.1049e-05	1.0037	0.12605	0.084648	0.0056693	0.069974	3.9738e-06
'fake'	2.7936e-05	35.907	93.669	0.00025209	1.0053	0.1785	0.12786	0.0051504	0.065515	2.641e-05
'fake'	5.6548e-05	38.105	90.607	0.00036697	1.0032	0.15674	0.12984	0.0048454	0.021402	5.0944e-05
'fake'	6.7866e-05	37.902	89.814	0.00044861	1.0033	0.18731	0.15286	0.0042187	0.012336	6.6474e-05
'fake'	7.3403e-05	37.704	89.474	0.00042351	1.0034	0.14932	0.1192	0.0045607	0.010396	6.8378e-05
'fake'	7.9749e-05	37.393	89.114	0.00043581	1.0036	0.15527	0.12049	0.0050042	0.007759	7.4845e-05
'fake'	7.6885e-05	37.575	89.272	0.00045617	1.0034	0.15878	0.12016	0.0048253	0.010794	6.581e-05
'fake'	4.8946e-05	40.346	91.234	0.00051088	1.0022	0.14299	0.11298	0.002892	0.014089	4.6254e-05
'fake'	9.8088e-05	35.493	88.215	0.00074017	1.0055	0.20167	0.14331	0.0051696	0.016988	0.000106
'fake'	8.1008e-05	36.243	89.046	0.00061491	1.0047	0.20167	0.155	0.0045605	0.020919	8.3146e-05
'fake'	8.1442e-05	35.345	89.022	0.00057141	1.0055	0.20167	0.14201	0.0056047	0.019659	8.2103e-05
'fake'	8.8566e-05	35.937	88.658	0.00054823	1.0047	0.20167	0.16182	0.0053674	0.021966	8.7569e-05
'fake'	8.3372e-05	34.967	88.921	0.00058639	1.0059	0.23527	0.15956	0.0058607	0.038422	8.4007e-05
'fake'	0.00010885	35.253	87.763	0.00058799	1.0054	0.20167	0.15927	0.006157	0.032198	0.00010958
'fake'	7.677e-05	37.434	89.279	0.00060234	1.0036	0.20167	0.14072	7.677e-05	0.049049	8.2915e-05
'fake'	9.0564e-05	35.421	88.561	0.00059849	1.0053	0.20167	0.13973	9.0564e-05	0.017828	9.4674e-05
'fake'	9.9591e-05	35.597	88.149	0.0006201	1.0049	0.2143	0.1699	9.9591e-05	0.0075677	9.9857e-05
'fake'	8.6868e-05	36.952	88.742	0.0006171	1.0038	0.20232	0.1873	8.6868e-05	0.036186	9.3703e-05
'fake'	0.00010953	35.276	87.736	0.00069352	1.0053	0.22616	0.2081	0.00010953	0.029545	0.00012021
'fake'	6.9372e-05	37.208	89.719	0.00054926	1.0036	0.20202	0.17229	6.9372e-05	0.020216	7.1426e-05
'fake'	5.425e-05	36.63	90.787	0.00043978	1.0043	0.20167	0.1522	5.425e-05	0.030751	5.0339e-05
'fake'	6.24e-05	36.283	90.179	0.00044605	1.0043	0.20167	0.15503	6.24e-05	0.038663	5.4458e-05
'fake'	0.0001096	36.426	87.733	0.00071931	1.0042	0.20167	0.1646	0.0001096	0.012264	0.00011548
'fake'	7.7542e-05	37.349	89.235	0.00061522	1.0036	0.20167	0.14717	7.7542e-05	0.039177	8.141e-05
'fake'	6.115e-05	36.412	90.267	0.0004505	1.0044	0.20167	0.15111	6.115e-05	0.042231	6.4219e-05
'fake'	5.0384e-05	37.571	91.108	0.00037773	1.0033	0.20167	0.12825	5.0384e-05	0.031618	5.0417e-05
'fake'	5.7057e-05	34.223	90.568	0.00043462	1.0064	0.22484	0.13981	5.7057e-05	0.020968	5.3417e-05
'fake'	0.00010006	36.217	88.128	0.00071932	1.0044	0.20167	0.15737	0.00010006	0.027854	0.00010648
'fake'	9.0619e-05	37.175	88.559	0.000688	1.0037	0.20167	0.15703	9.0619e-05	0.011651	9.457e-05
'fake'	0.00010313	36.528	87.997	0.00078093	1.0042	0.22104	0.17036	0.004157	0.022727	0.00011146

Table 19 : Best-10 LDA result

LDA:

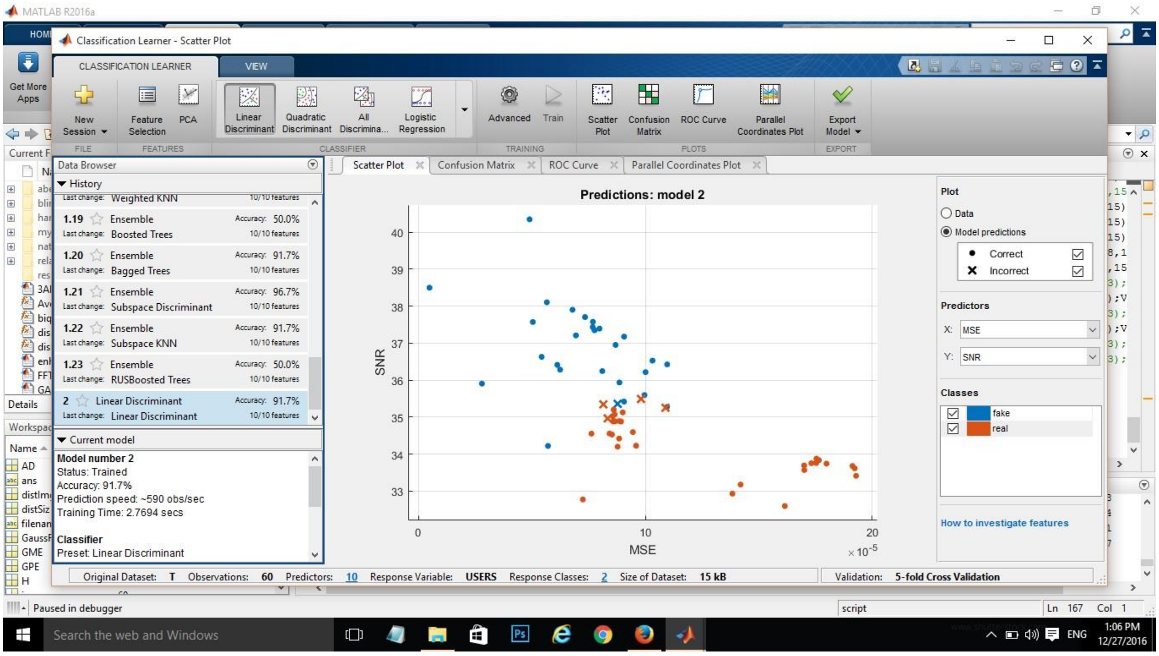


Figure C.30: Scatter plot

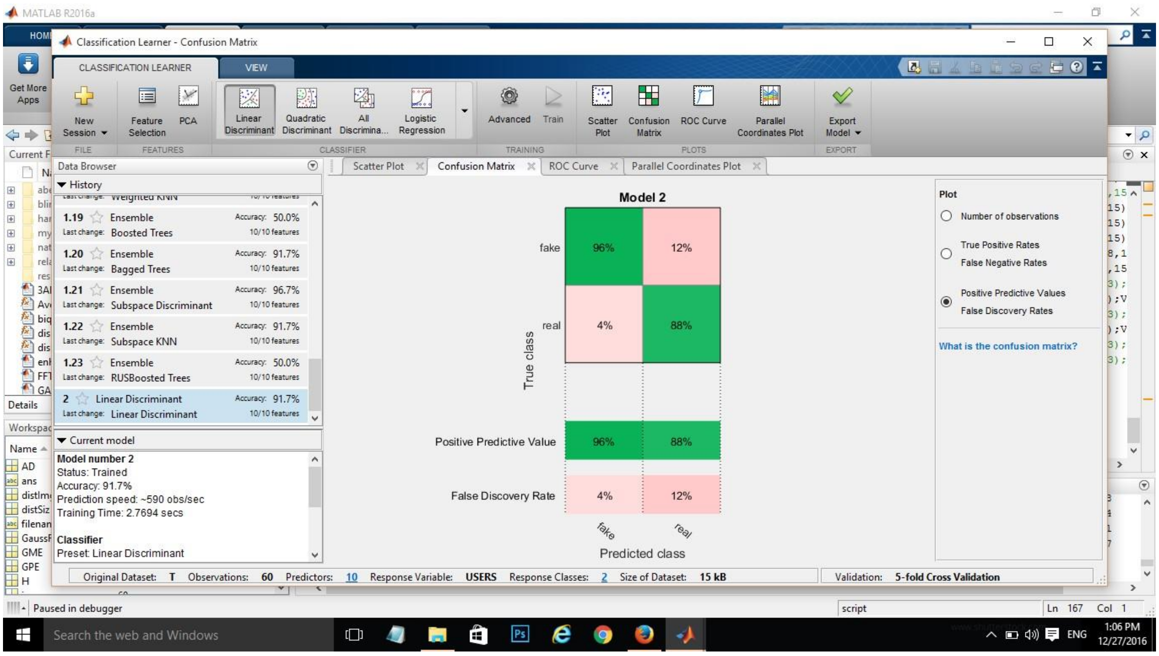


Figure C.31: Confusion matrix

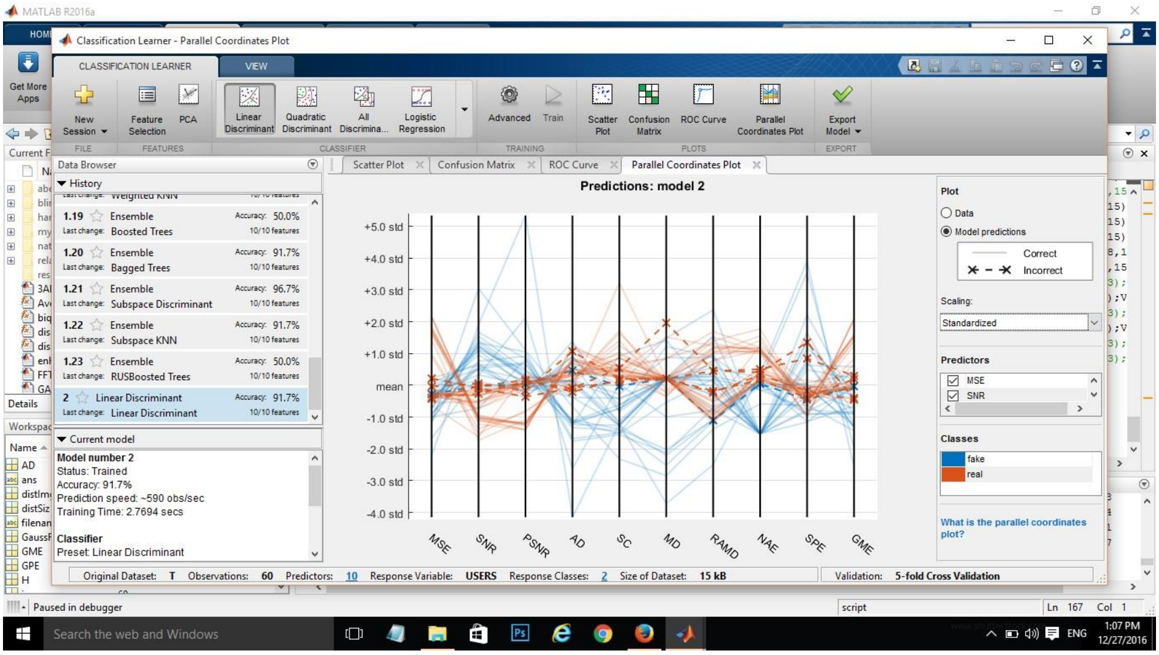


Figure C.32: Parallel coordinate

Best-10 LDA results plotted on scatter plot Figure C.30, confusion matrix

Figure C.31, parallel coordinate Figure C.32

QDA:

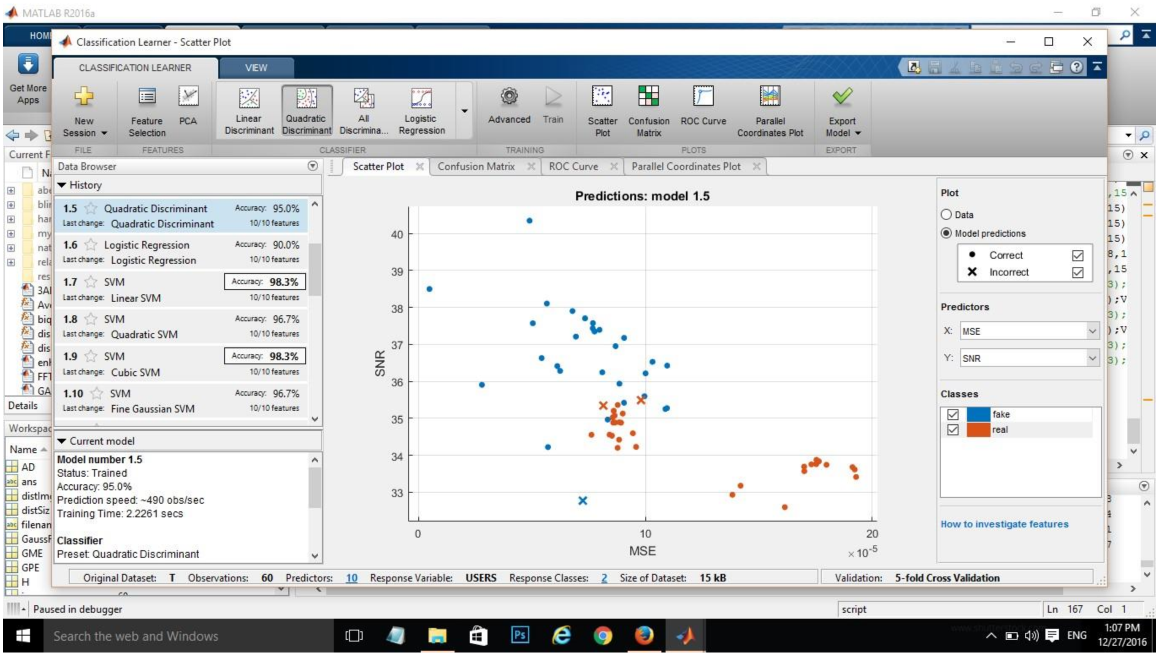


Figure C.33: Scatter plot

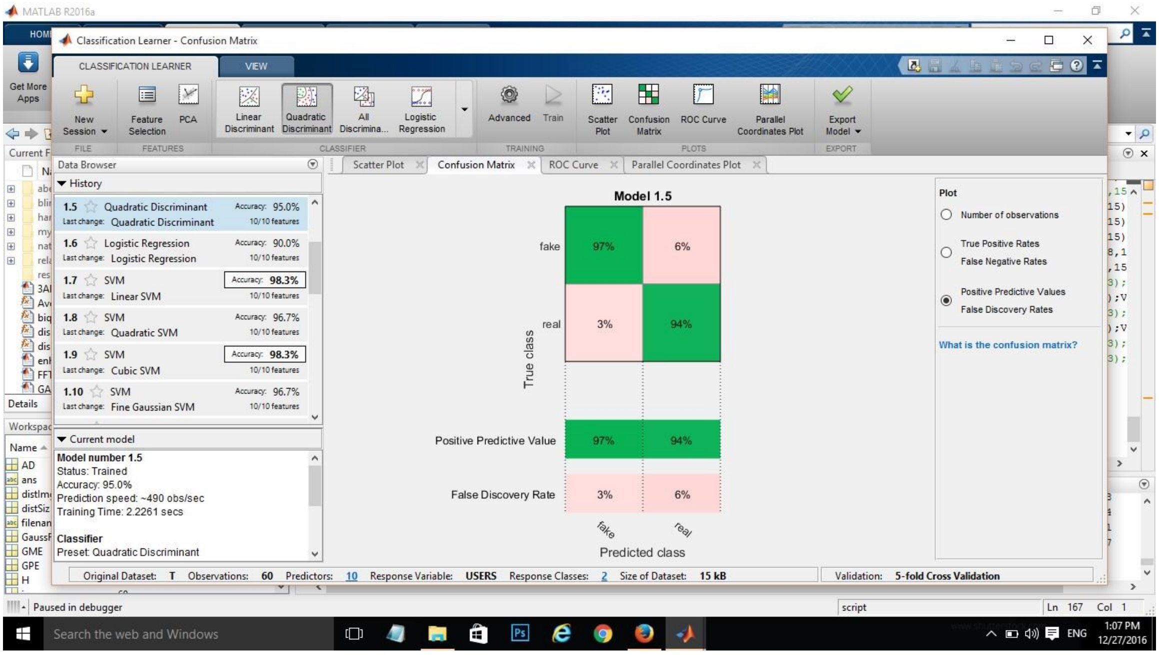


Figure C.34: Confusion matrix

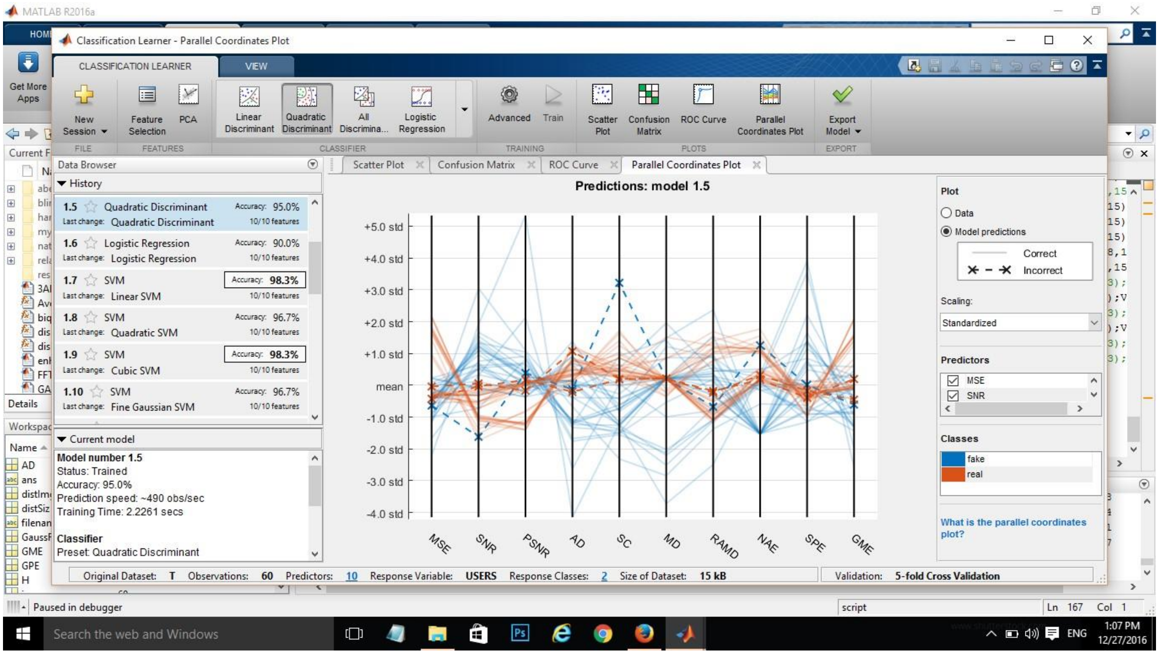


Figure C.35: Parallel coordinate

Best-10 QDA results plotted on scatter plot Figure C.33, confusion matrix Figure C.34, parallel coordinate Figure C.35

LINEAR-SVM:

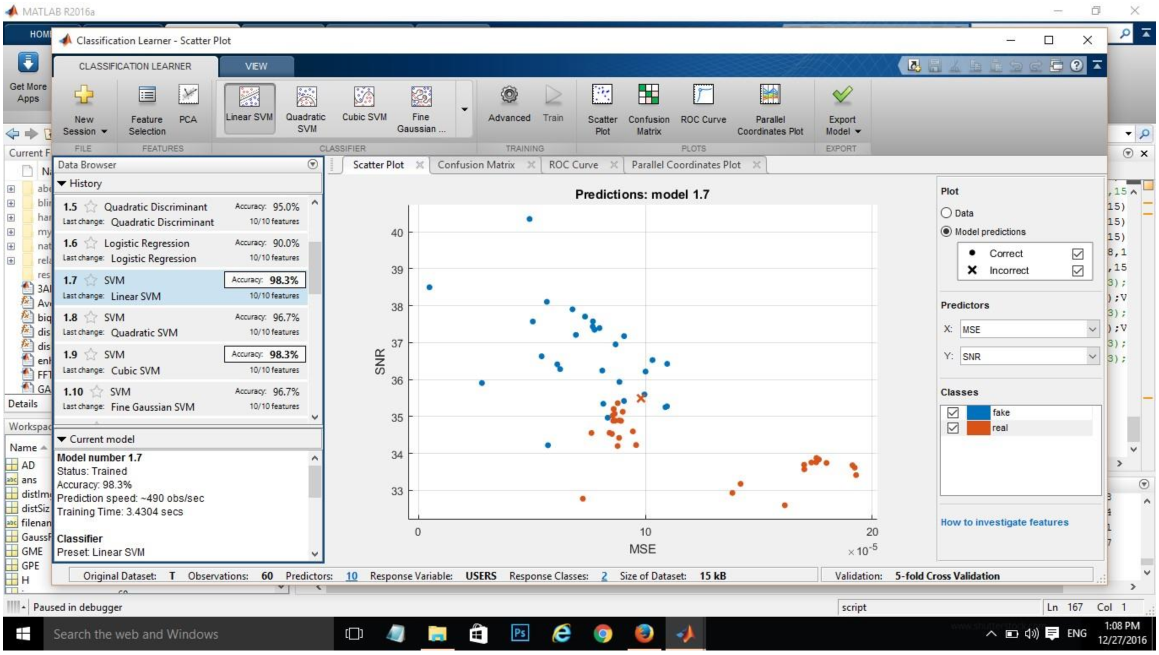


Figure C.36: Scatter plot

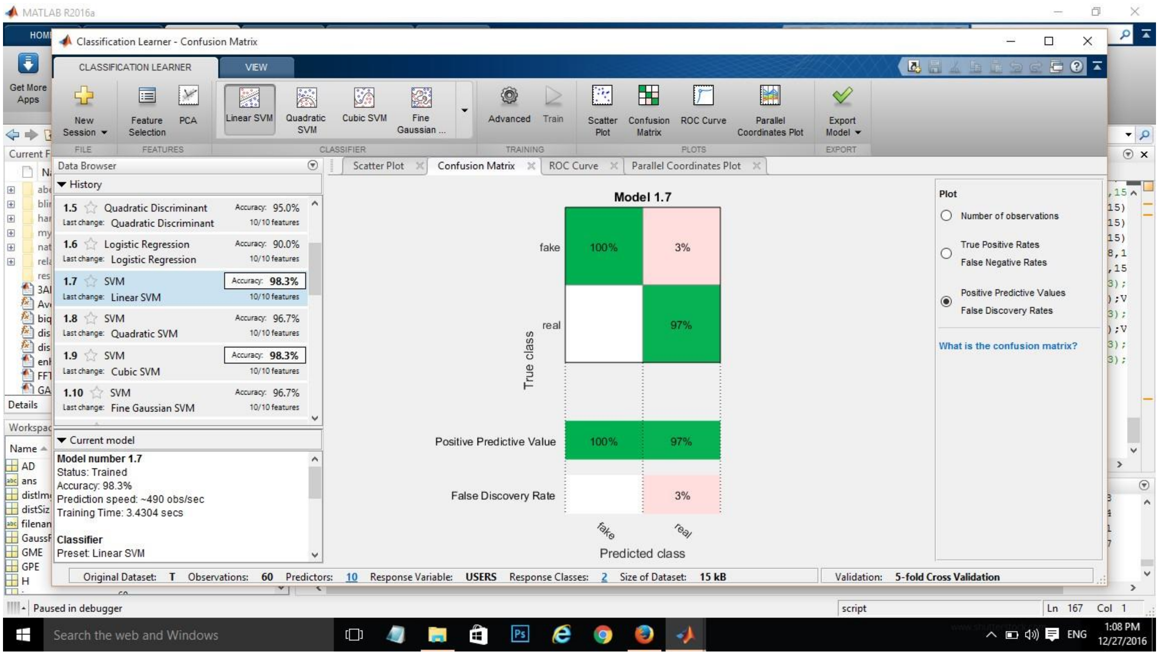


Figure C.37: Confusion matrix

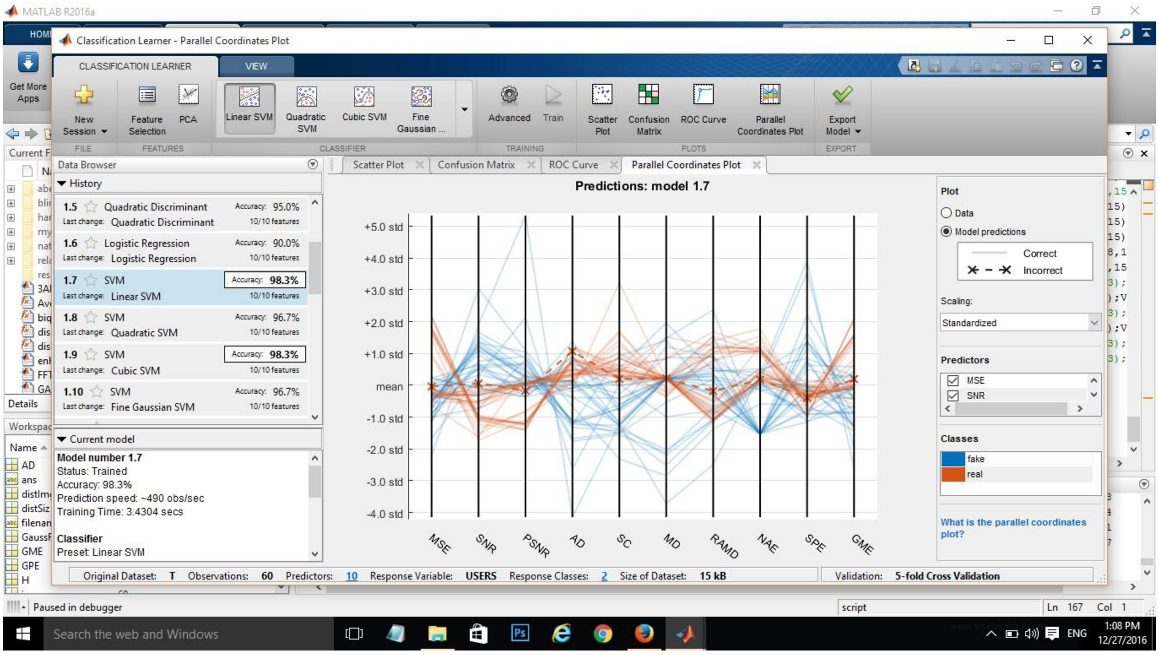


Figure C.38: Parallel coordinate

Best-10 Linear SVM results plotted on, scatter plot Figure C.36, confusion matrix Figure C.37, parallel coordinate Figure C.38

QUADRATIC SVM:

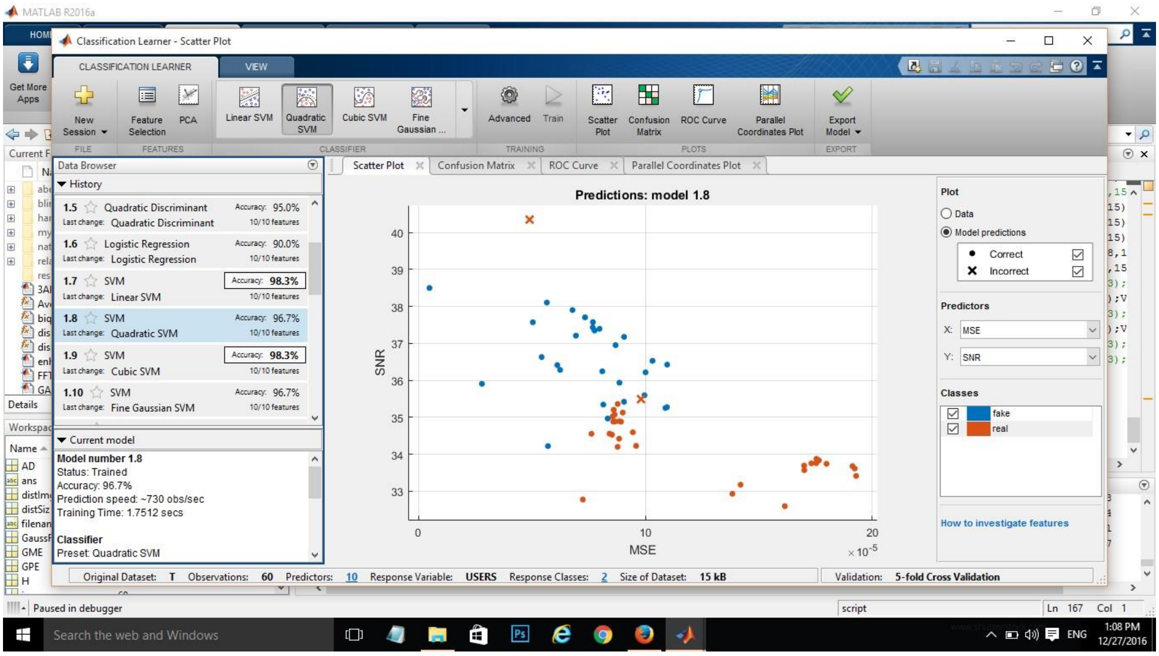


Figure C.39: Scatter plot

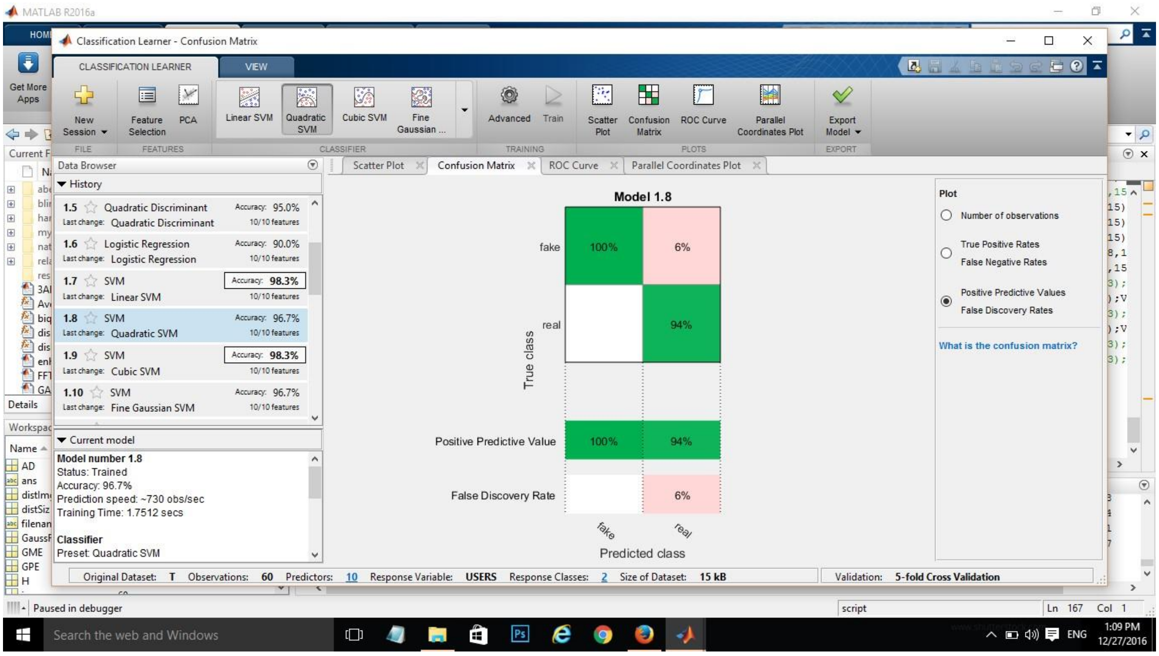


Figure C.40: Confusion matrix

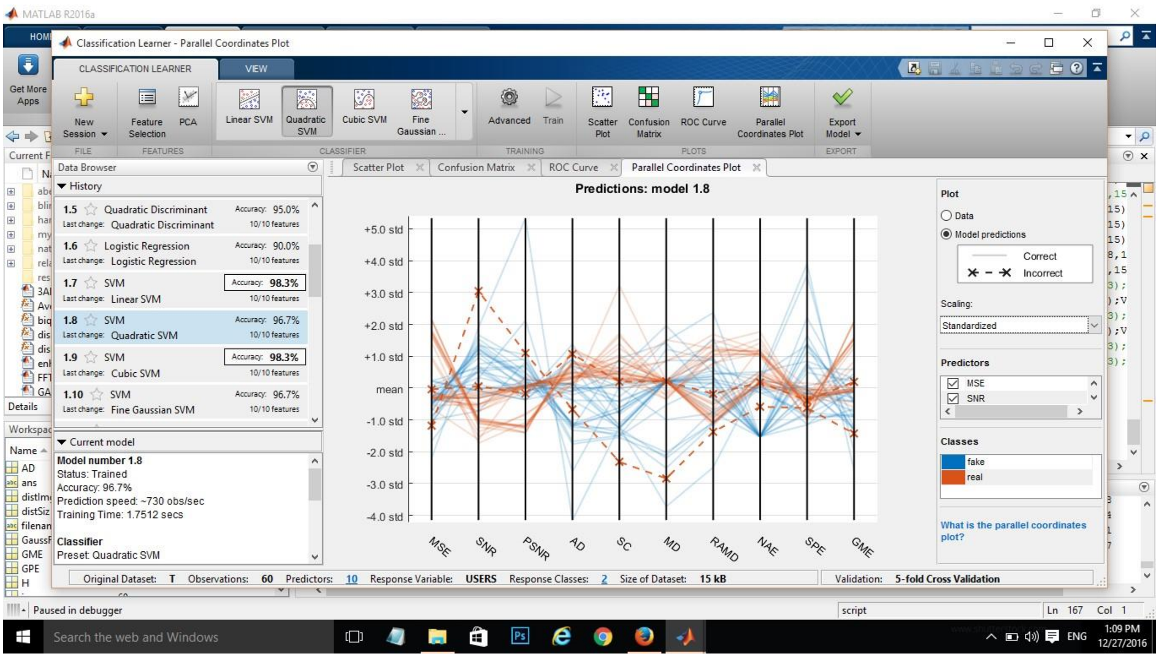


Figure C.41: Parallel coordinate

Best-10 Quadratic SVM results plotted on, scatter plot Figure C.39, confusion matrix Figure C.40, parallel coordinate Figure C.41

LOGISTIC REGRESSION:

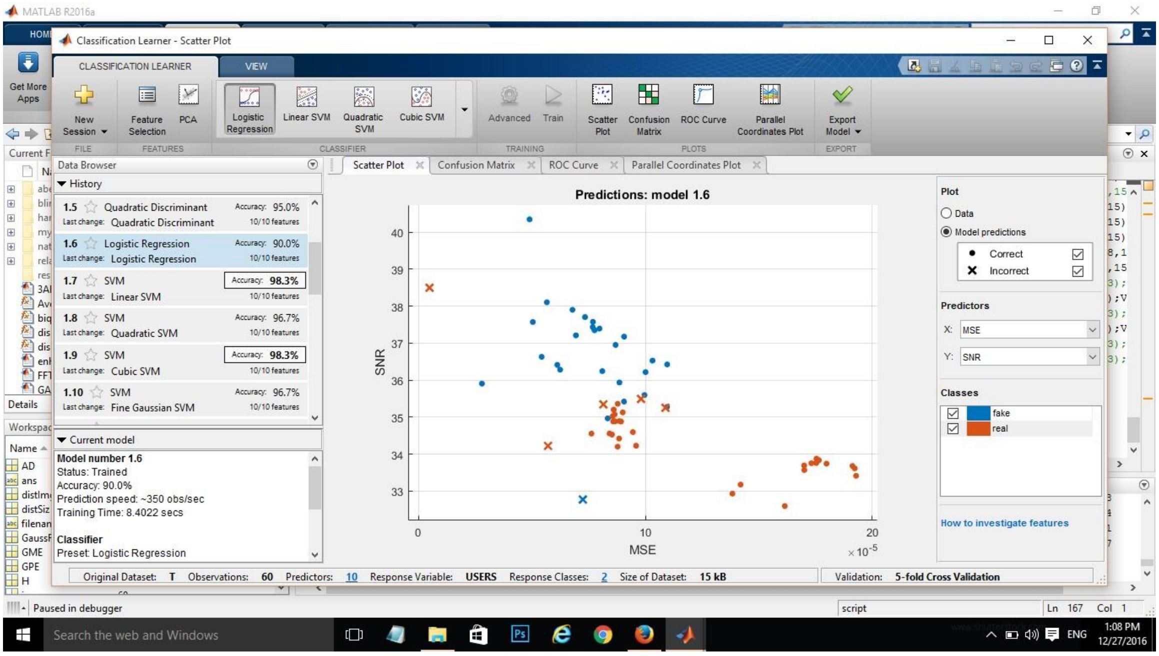


Figure C.42: Scatter plot

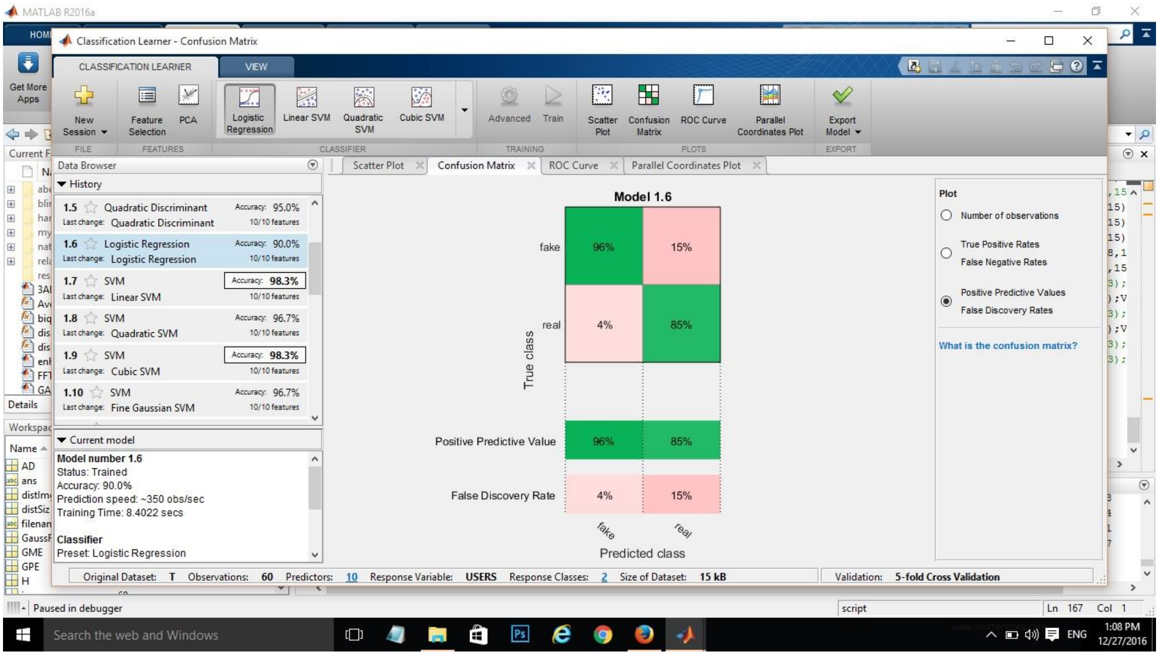


Figure C.43: Confusion matrix

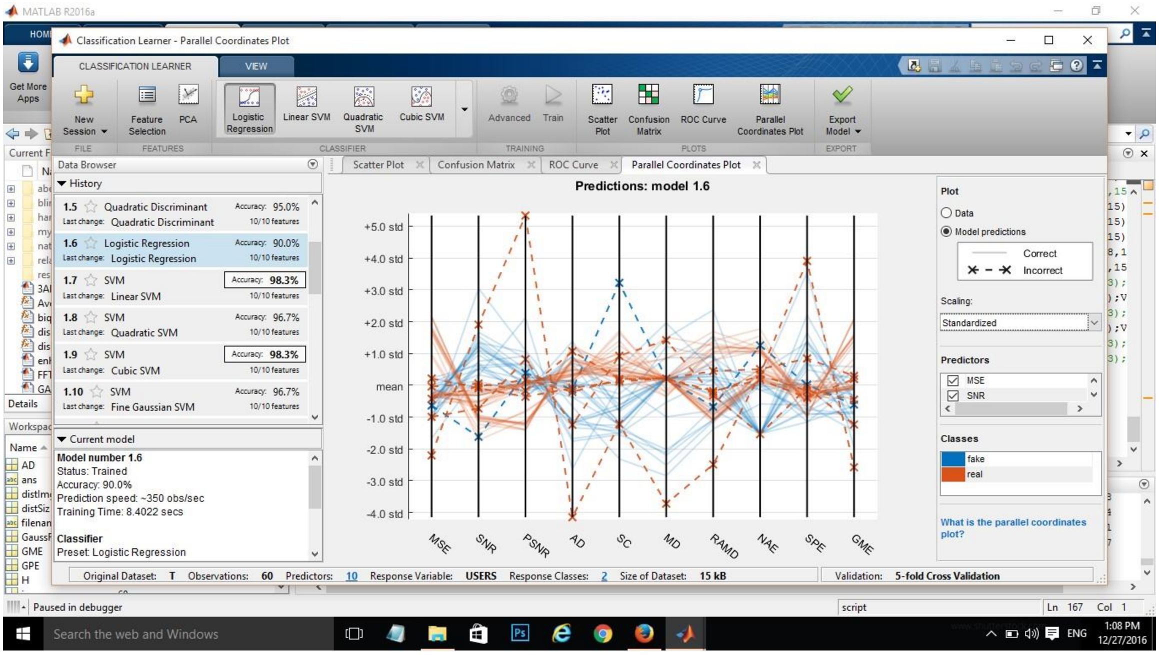


Figure C.44: Parallel coordinate

Best-10 logistic regression results plotted on, scatter plot Figure C.42, confusion matrix Figure C.43, parallel coordinate Figure C.44

Users	Real-min	Real-max	Fake-min	Fake-max
MSE	0.000138	0.000195	2.793	**7.974**
SNR	31.738	33.986	35.907	**40.288**
PSNR	85.153	86.720	89.114	**101.31**
NCC	0.9953	0.9970	0.9972	**0.9998**
AD	0.0006	0.0006	0.0002	**0.0005**
SC	1.0055	1.0087	1.0021	**1.0053**
MD	0.190	0.238	0.126	**0.187**
RAMD	0.161	0.197	0.100	**0.152**
NAE	0.007	0.009	0.002	**0.005**
TED	0.002	0.003	0.001	**0.003**
SPE	0.0077	0.0176	0.0061	**0.0173**
SME	19.643	29.901	3.040	**10.933**
GME	0.00012	0.00018	2.641	**6.872**
GPE	0.188	0.233	0.100	**0.174**
TCD	0	0.087	0	**0.082**

Table C.4: Min, max of 60 users

Table C.4 Minimum and maximum of each feature for 60 users in table C.1

Appendix D: Screenshots of classification results

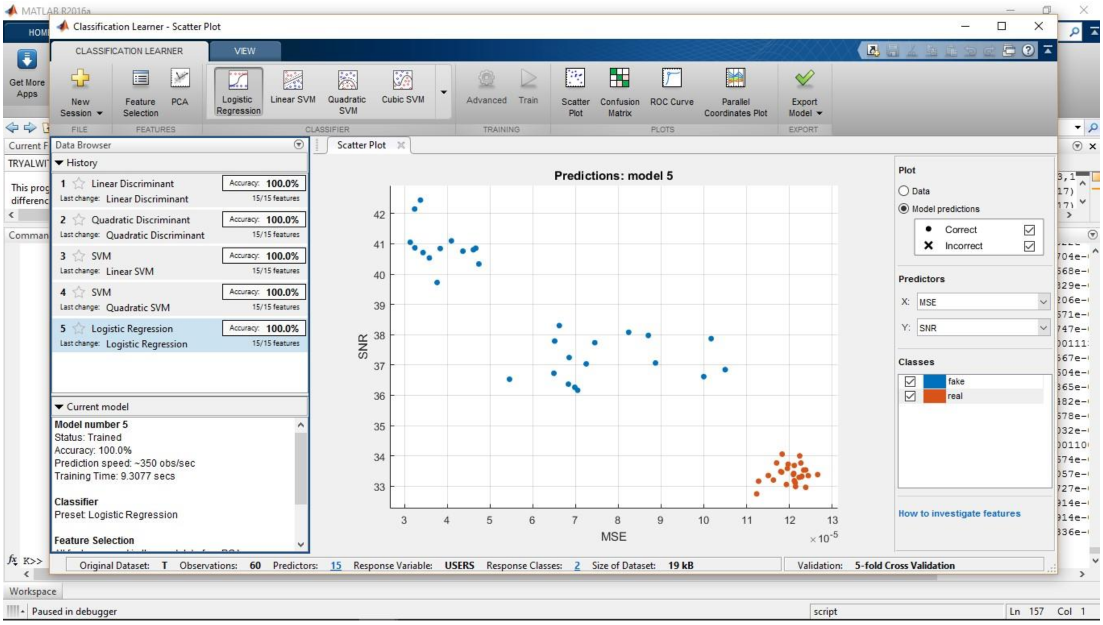


Figure D.1: Training model

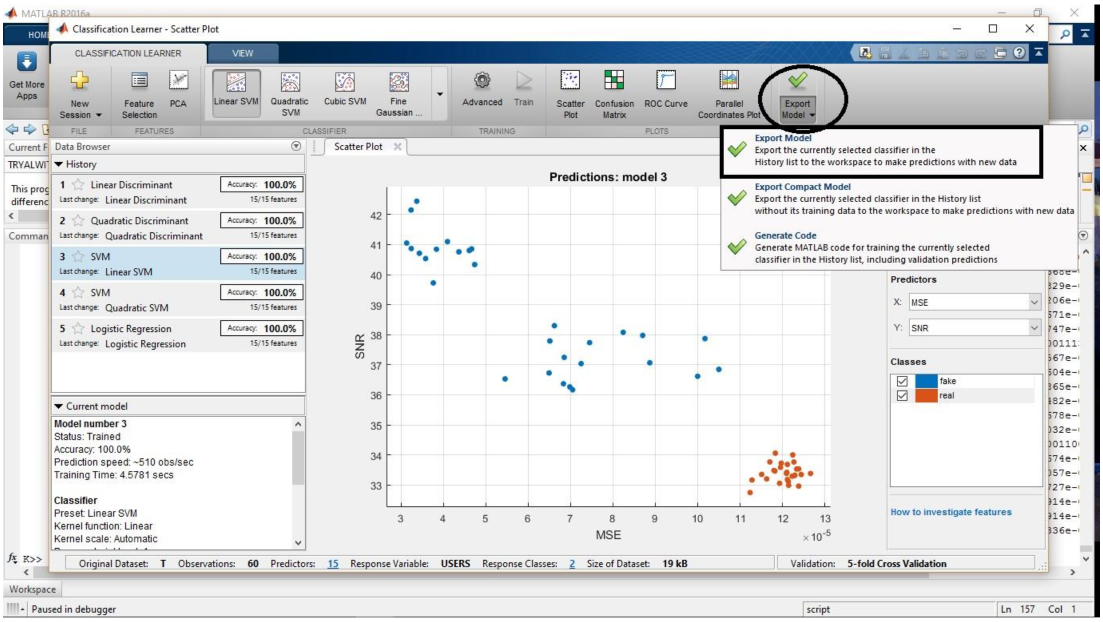


Figure D.2: Exporting process

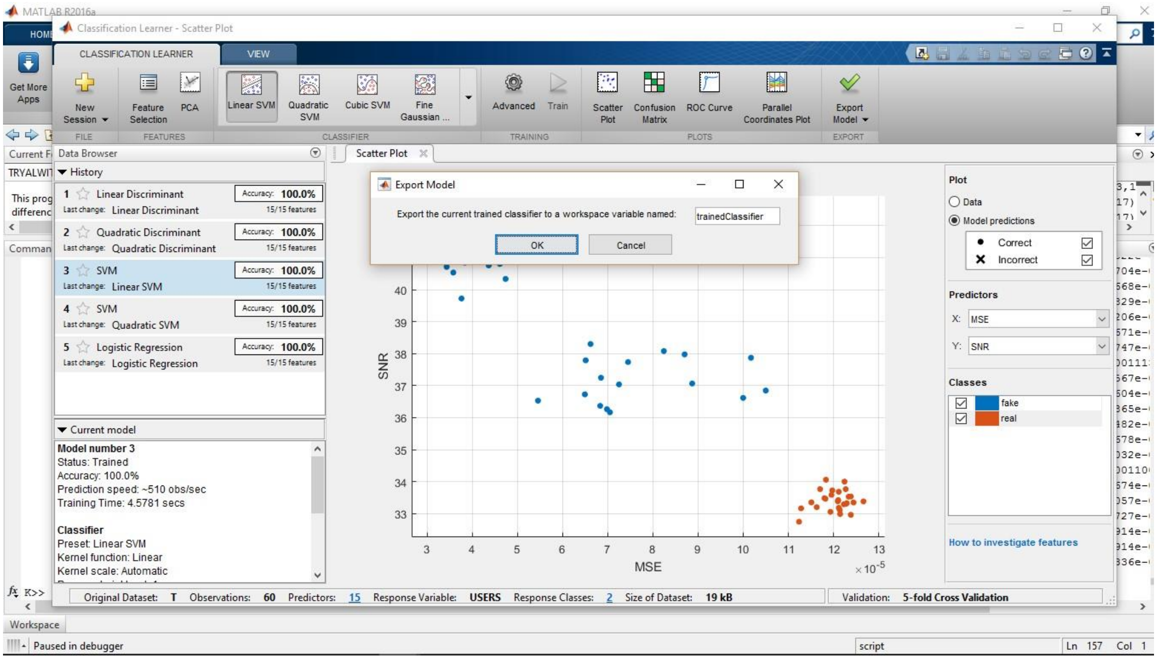


Figure D.3: Name the exporting model

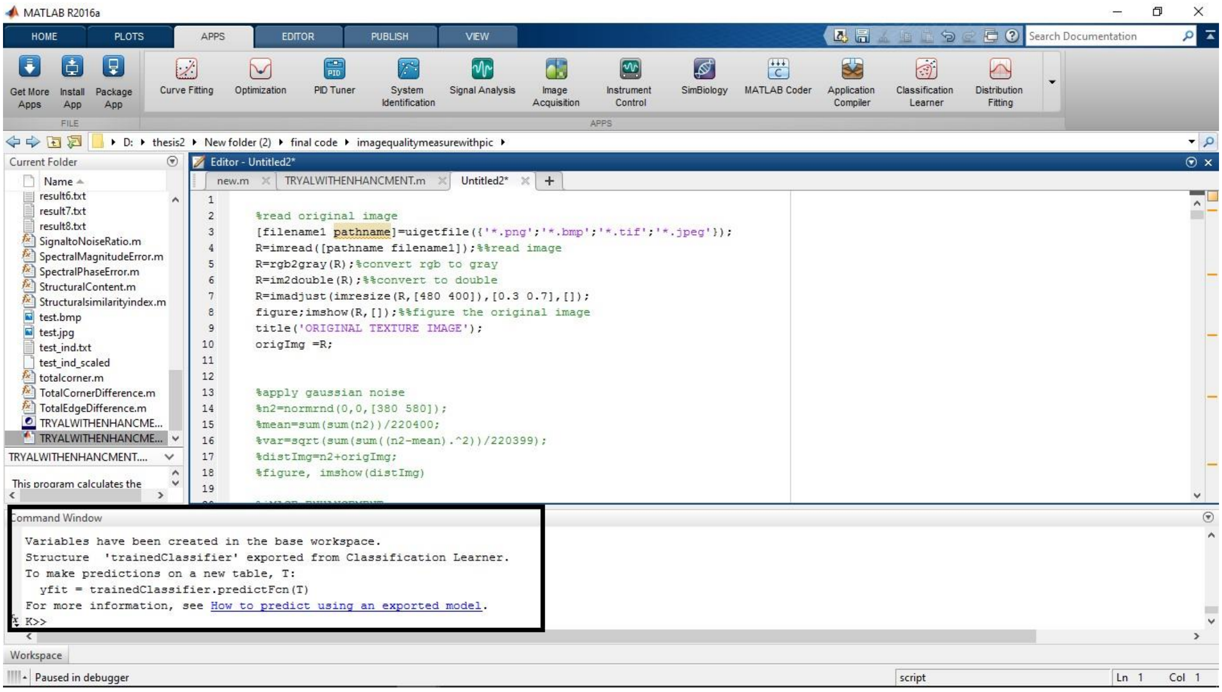


Figure D.4: Exported training model

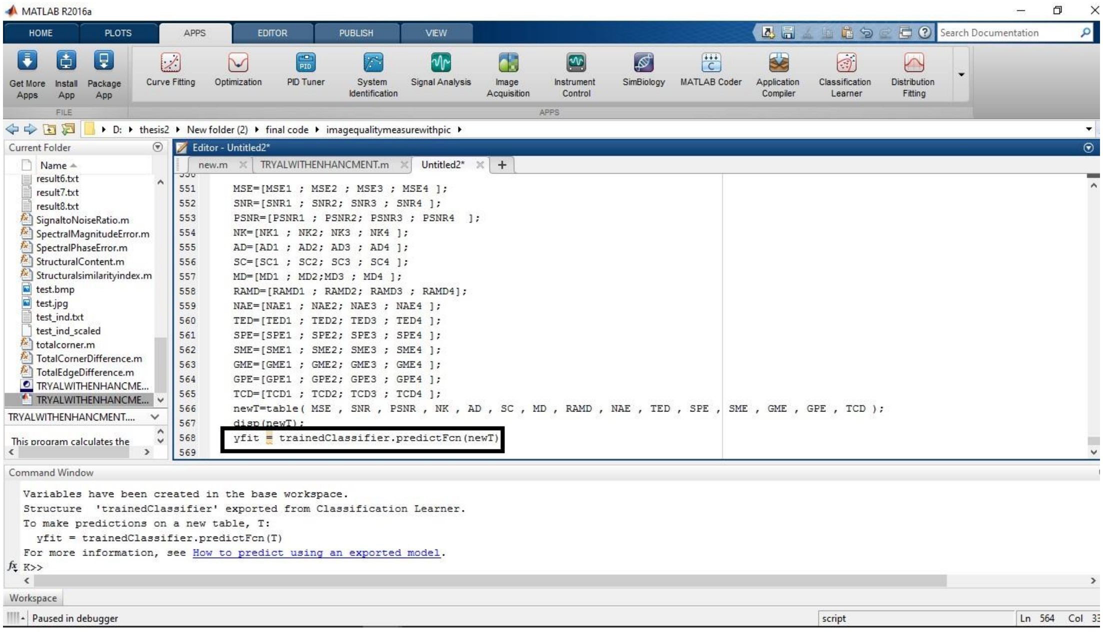


Figure D.5: Code and function for classification

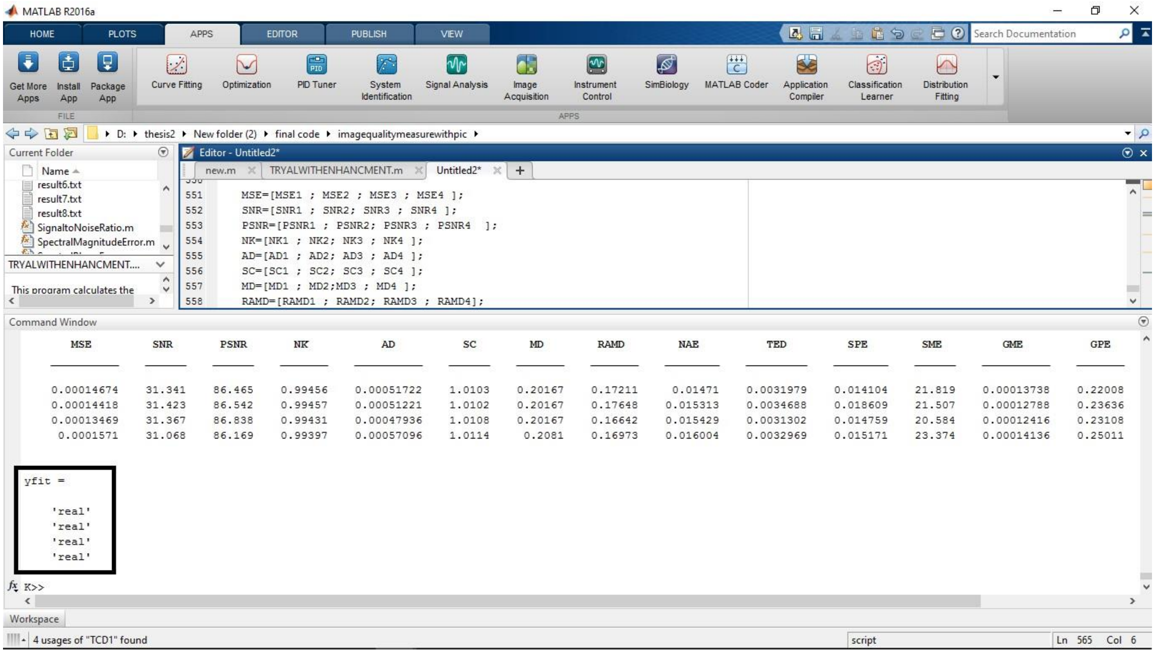


Figure D.6: Table of four images for classification

Appendix E: Screenshots of experimental results with Gaussian noise [3.2]

For real image: With variance 0 and mean 0: Screenshot of Original Image and Distortion with Mean 0 and Variance 0:

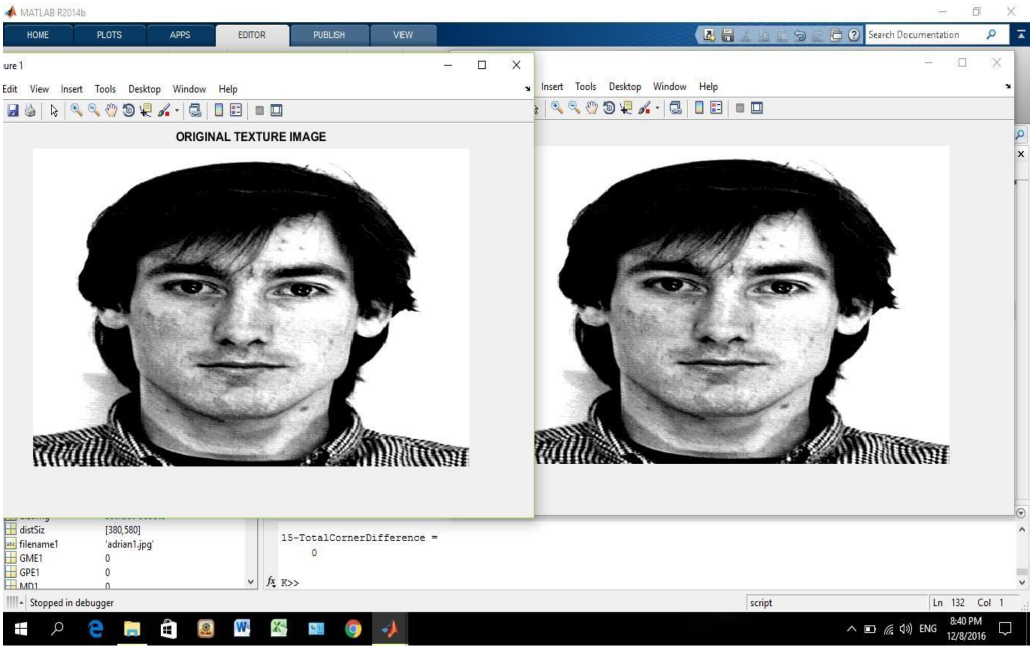


Figure: E.1: Real and distorted image using Gaussian noise

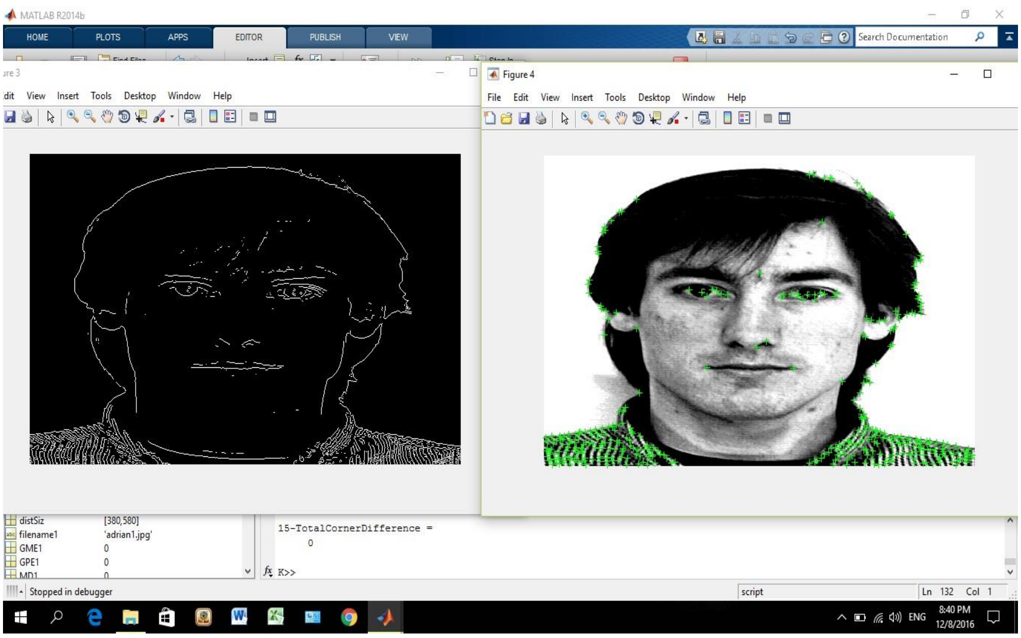


Figure: E.2: Screenshot of edge and corner detection of image

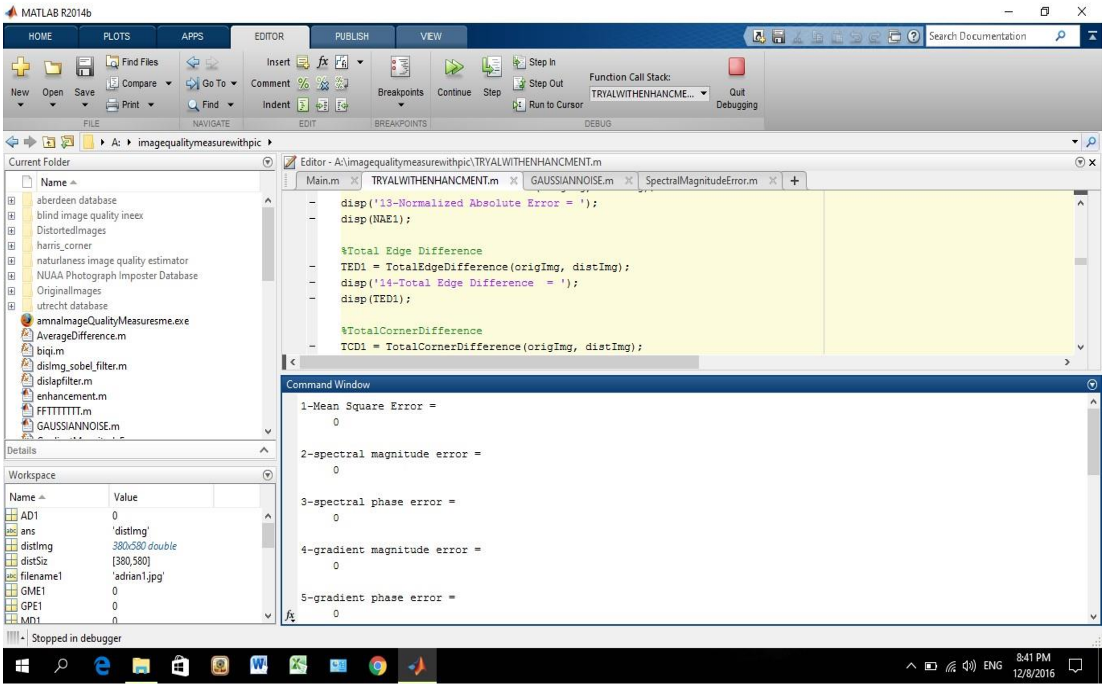


Figure: E.3: Results obtained from the present experiment on Gaussian noise, mean 0, variance 0

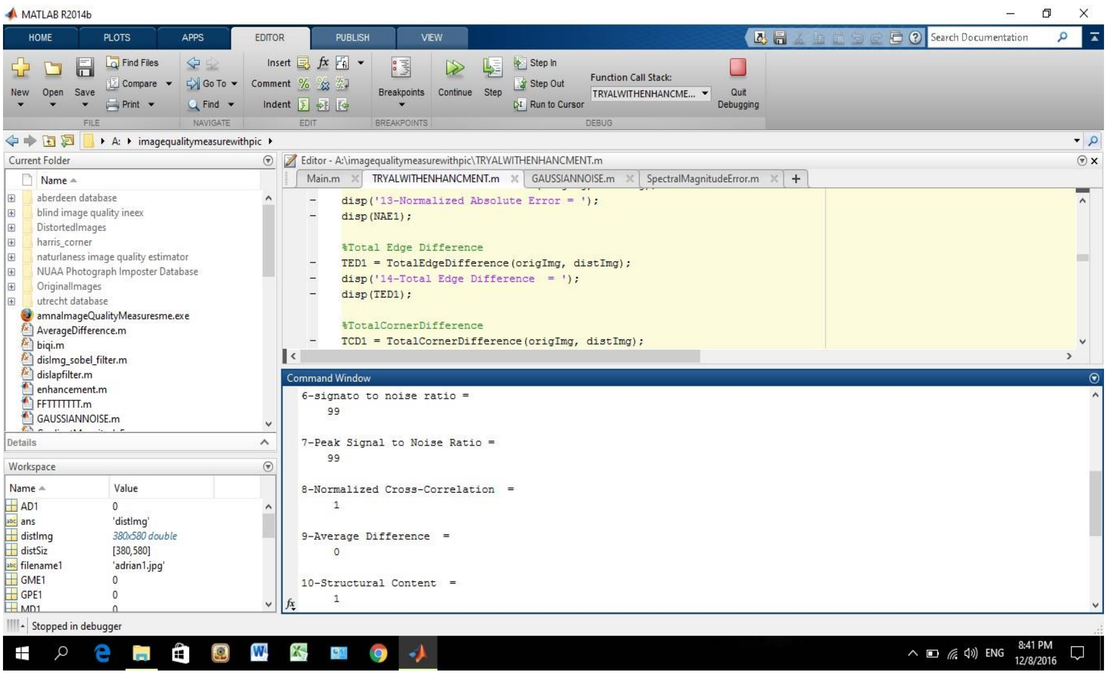


Figure: E.4: Results obtained from the present experiment on Gaussian noise, mean 0, variance 0

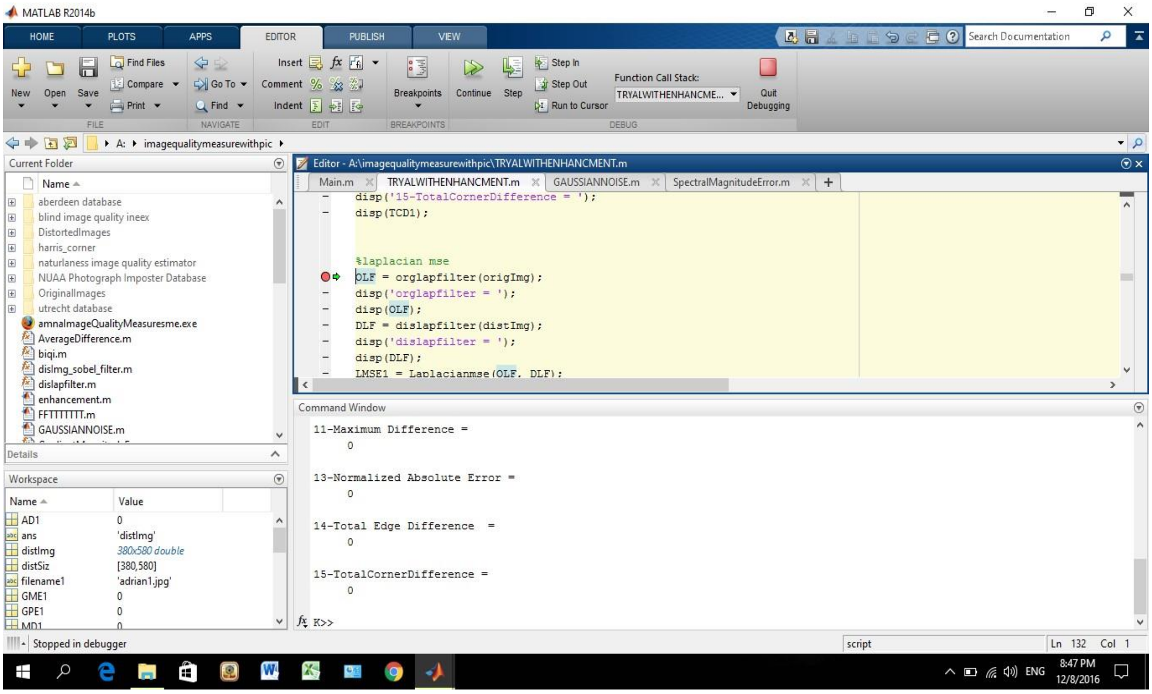


Figure: E.5: Results obtained from the present experiment on Gaussian noise, mean 0, variance 0

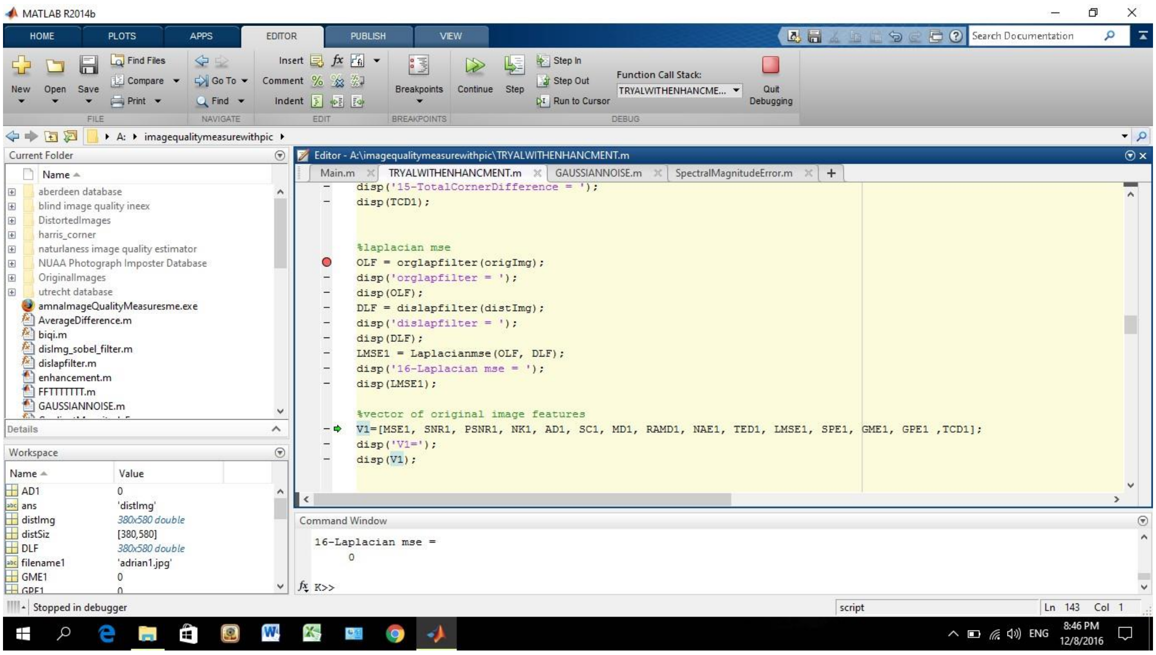


Figure: E.6: Results obtained from the present experiment on Gaussian noise, mean 0, variance 0

Experiment with Gaussian noise (variance 0 and mean 0) shows the real and distorted image using Gaussian noise Figure: E.1, represents Screenshot of Edge and

Corner Detection of Image Figure: E.2, show the results obtained from the present experiment on Gaussian noise, mean 0, variance 0. Figure: E.3, 4, 5, 6

Screenshot of Original Image and with Gaussian Noise with Variance 0.025 and mean 0:

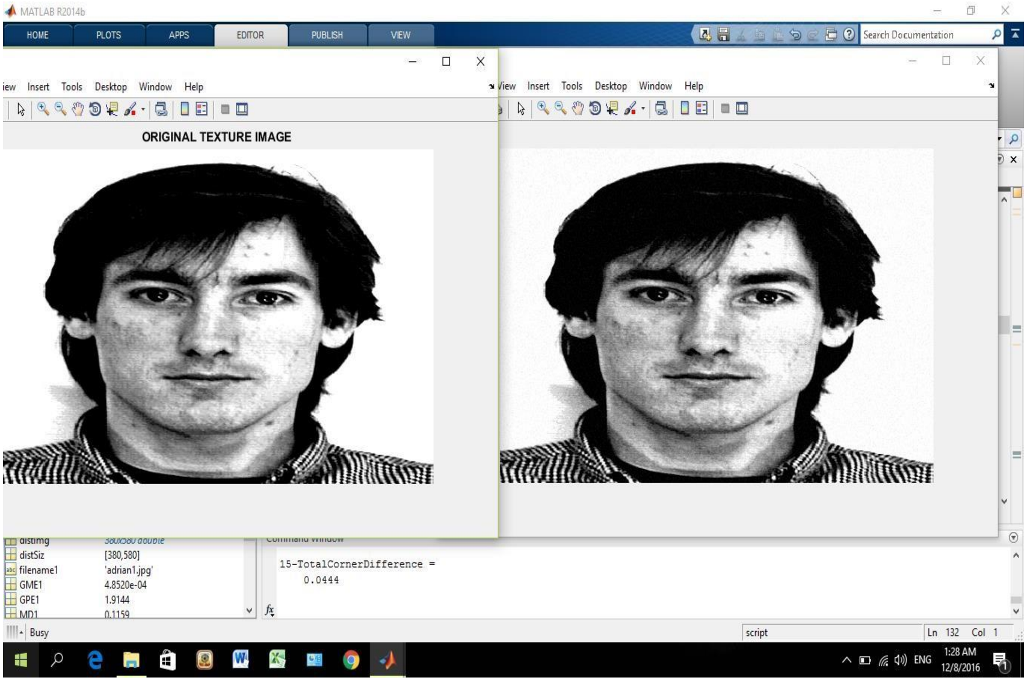


Figure: E.7: Real and distorted image using Gaussian noise

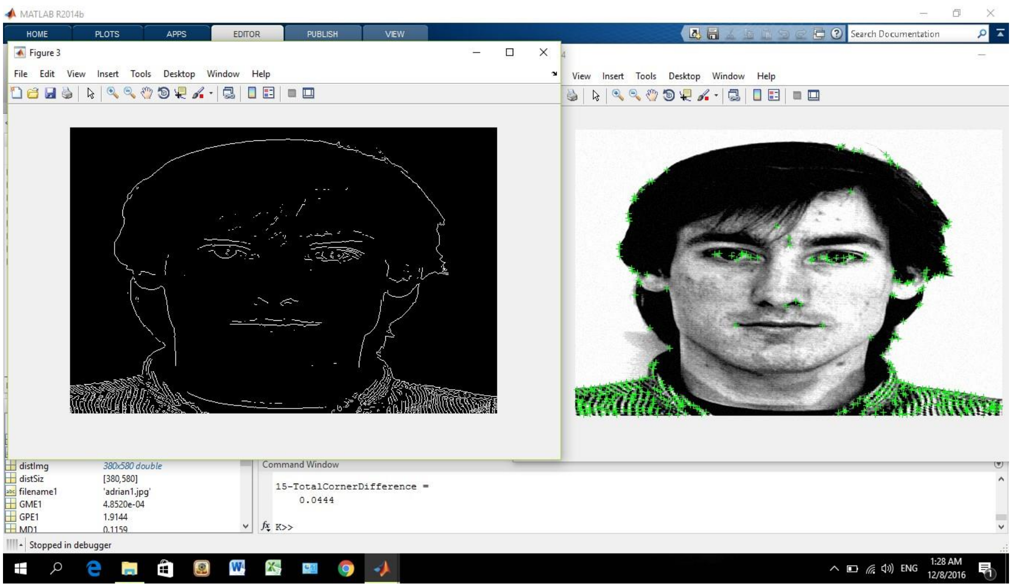


Figure: E.8: Represents Screenshot of edge and corner detection of image

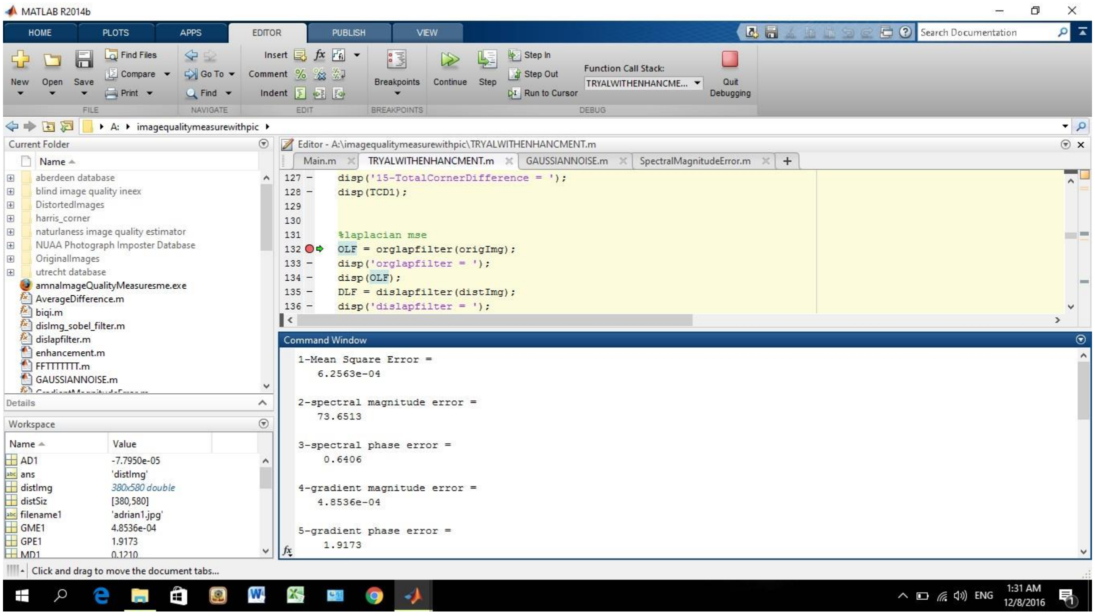


Figure: E.9: Results obtained from the present experiment on Gaussian noise, mean 0, variance 0.025

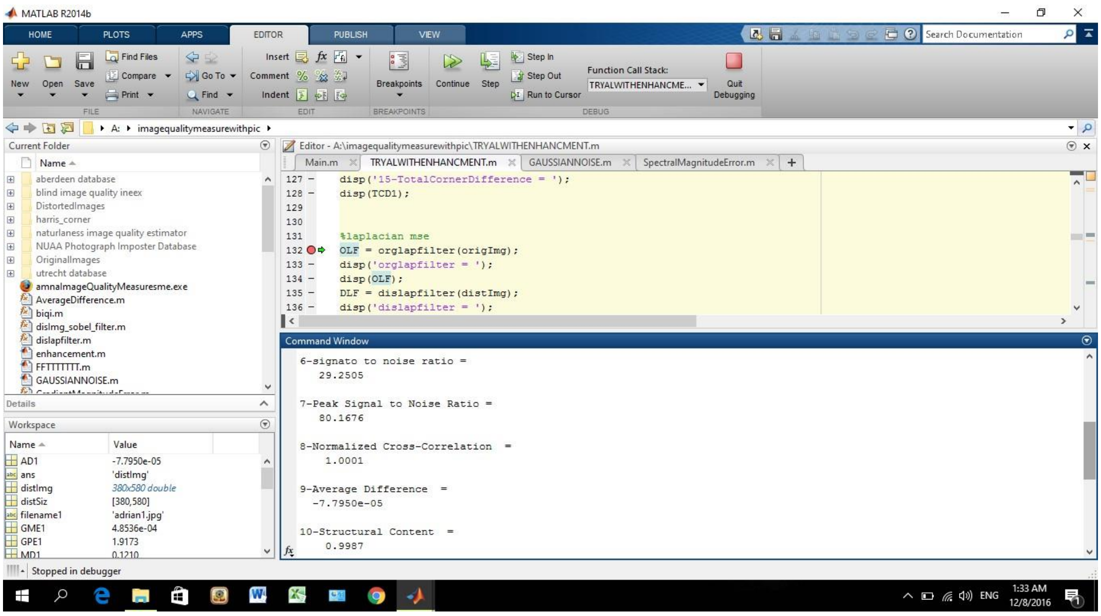


Figure: E.10: Results obtained from the present experiment on Gaussian noise, mean 0, variance 0.025

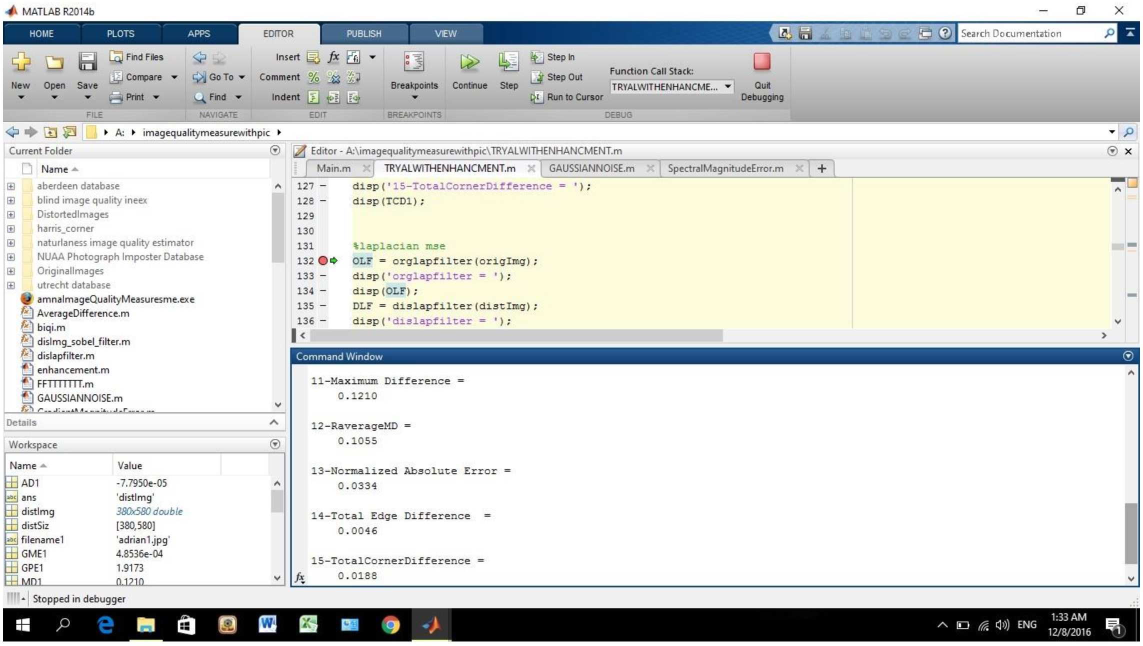


Figure E.11: Results obtained from the present experiment on Gaussian noise, mean 0, variance 0.025

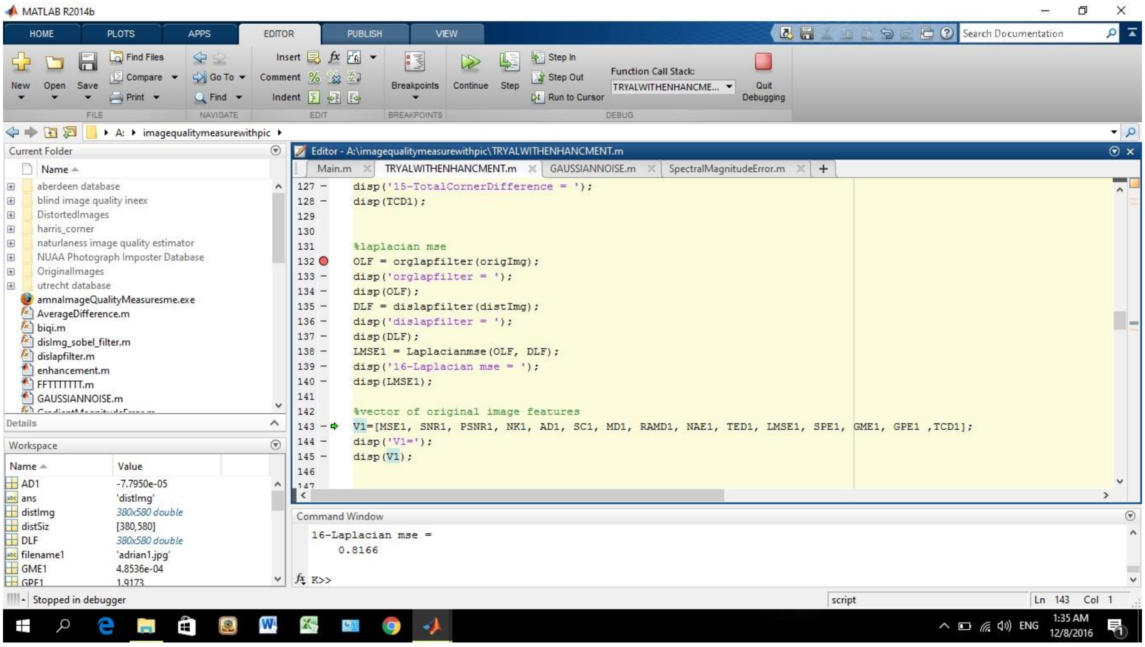


Figure: E.12: results obtained from the present experiment on Gaussian noise, mean 0, variance 0.025

Experiment with Gaussian noise (variance 0 and mean 0) shows the real and distorted image using Gaussian noise Figure: E.7, represents Screenshot Of Edge And Corner Detection Of Image Figure: E.8, show the results obtained from the present experiment on Gaussian noise, mean 0, variance 0.025 Figure: E.9, 10, 11, 12

Screenshot of Original Image and With Gaussian Noise with Variance 0.05 and Mean 0:

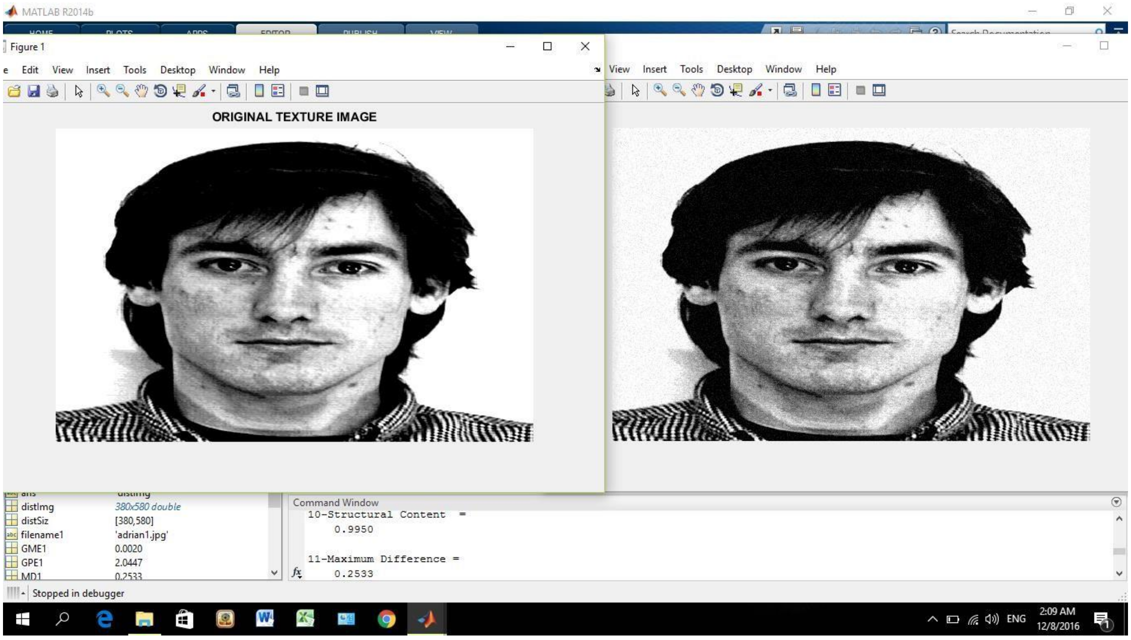


Figure: E.13: Real and distorted image using Gaussian noise

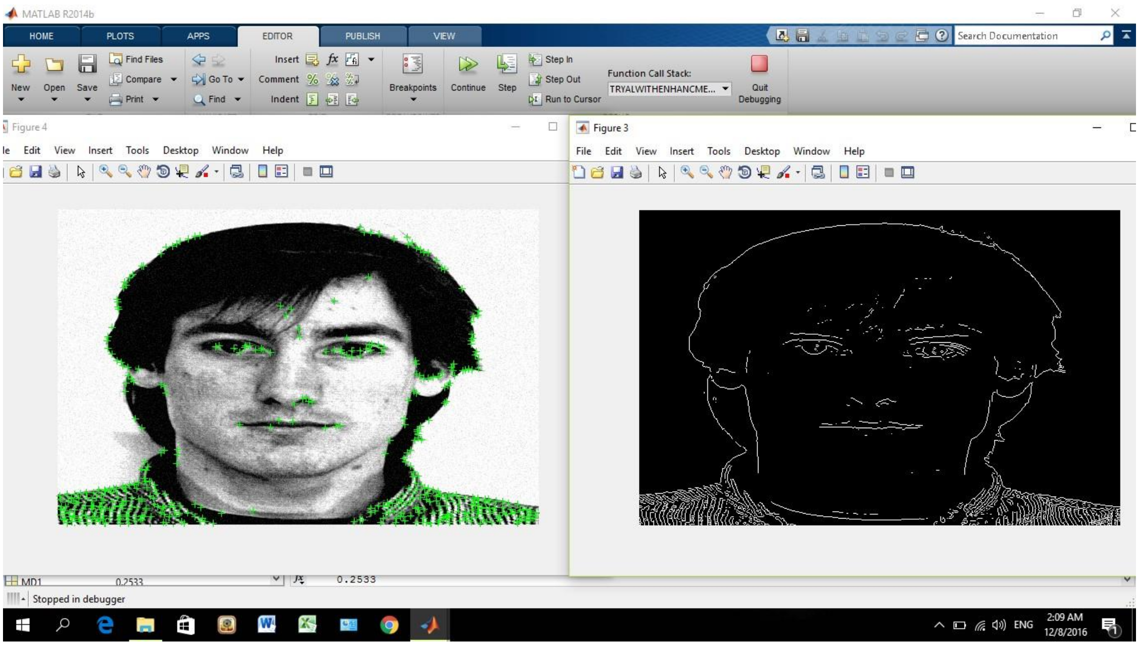


Figure: E.14: Represents screenshot of edge and corner detection of image

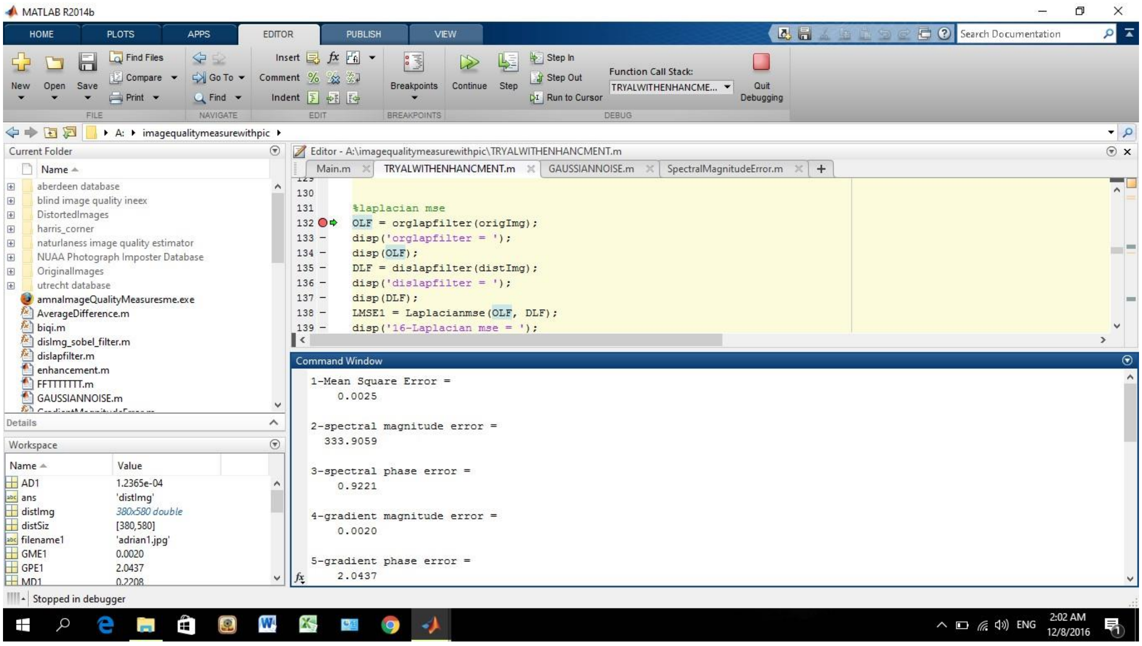


Figure: E.15: Results obtained from the present experiment on Gaussian noise, mean 0, variance 0.05

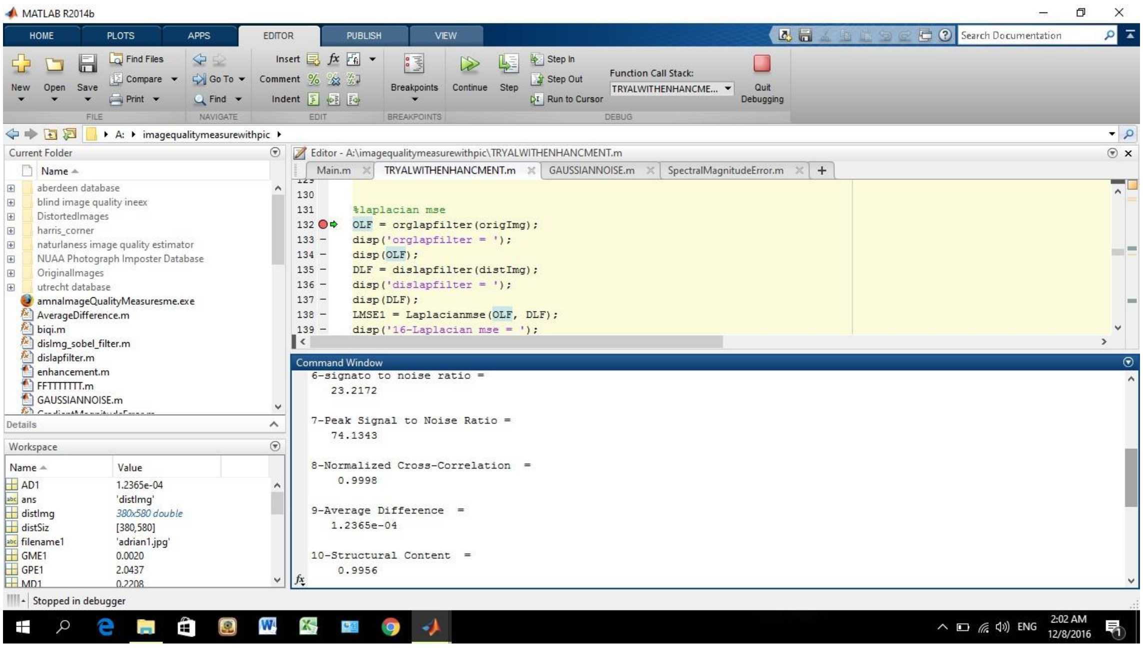


Figure: E.16: Results obtained from the present experiment on Gaussian noise, mean 0, variance 0.05

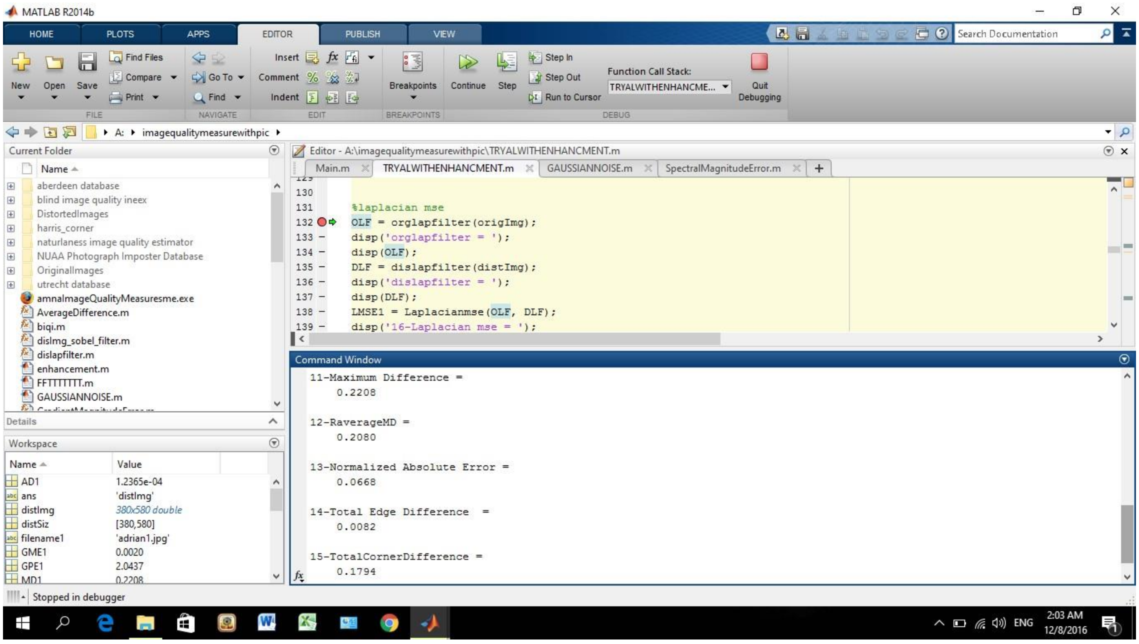


Figure: E.17: Results obtained from the present experiment on Gaussian noise, mean 0, variance 0.05

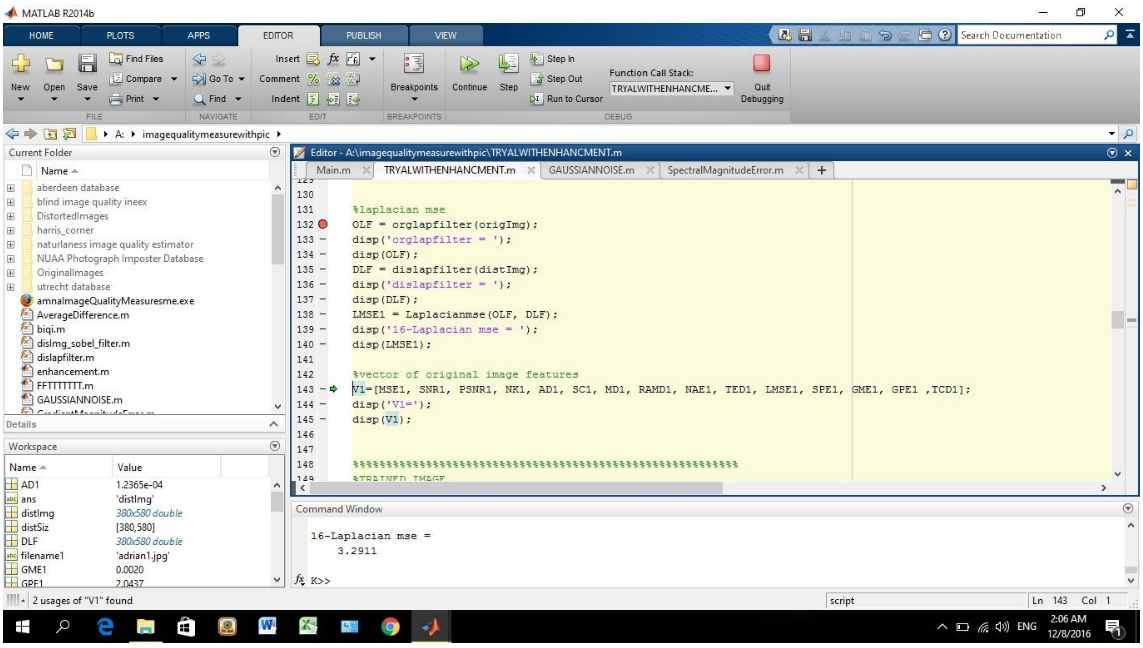


Figure: E.18: Results obtained from the present experiment on Gaussian noise, mean 0, variance 0.05

Experiment with Gaussian noise (variance 0 and mean 0) shows the real and distorted image using Gaussian noise Figure: E.13, represents Screenshot Of Edge And Corner Detection Of Image Figure: E.14, show the results obtained from the present experiment on Gaussian noise, mean 0, variance 0.05 Figure: E.15, 16, 17, 18

Screenshot of Original Image and With Gaussian Noise with Variance 0.1 and Mean 0:

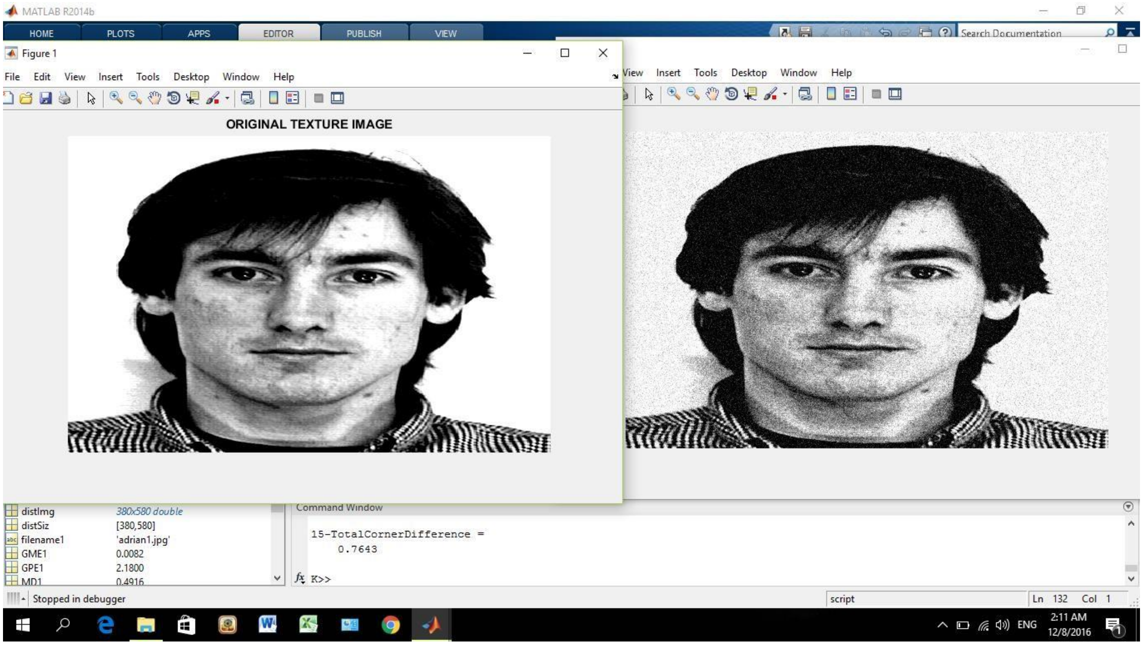


Figure: E.19: Real and distorted image using Gaussian noise

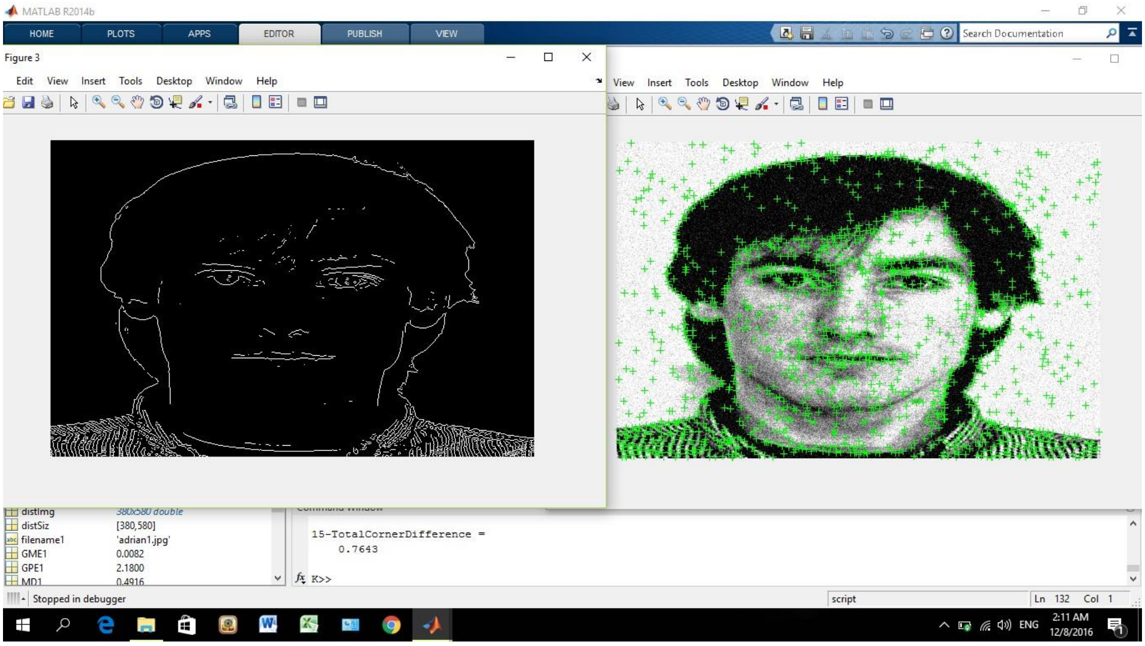


Figure: E.20: Represents screenshot of edge and corner detection of image

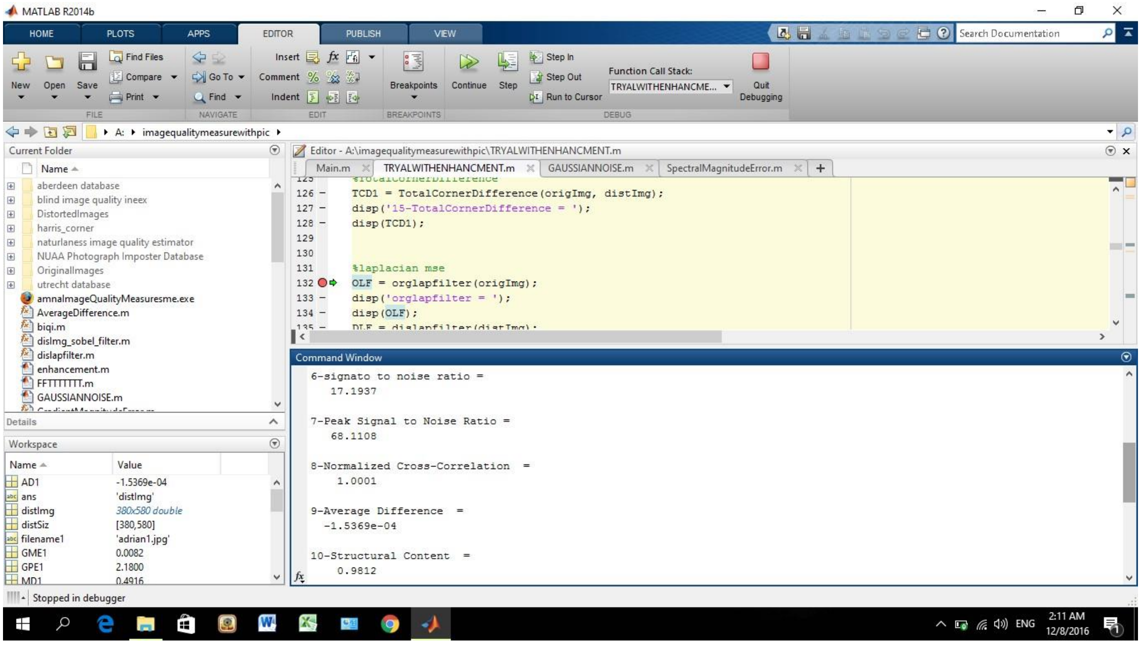


Figure: E.21: Results obtained from the present experiment on Gaussian noise, mean 0, variance 0.1

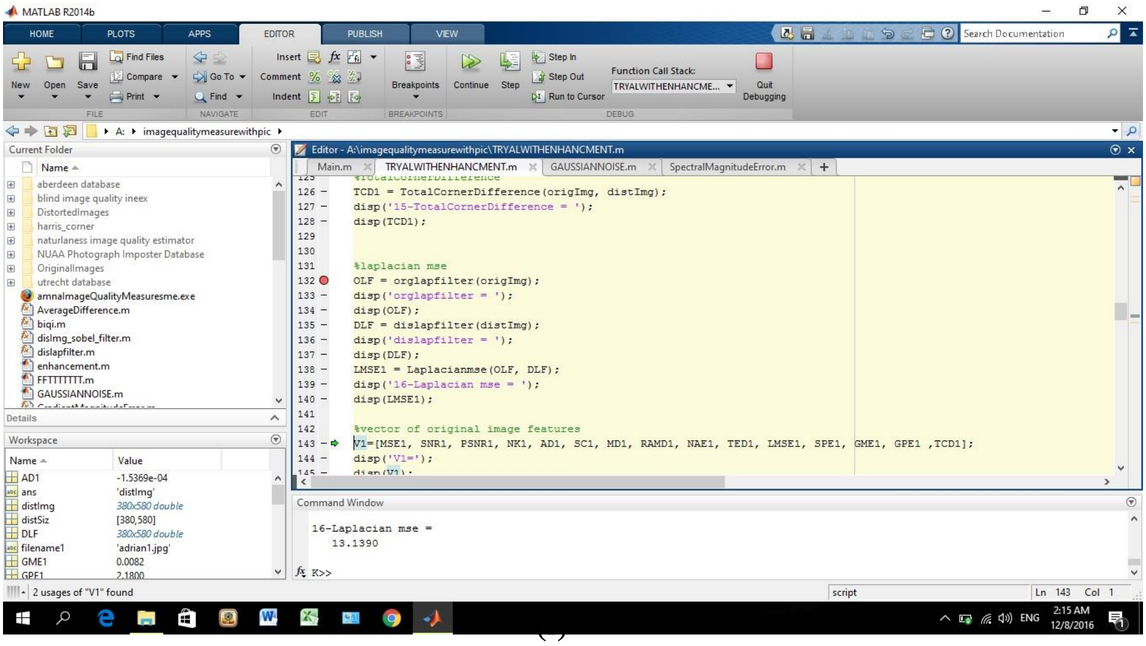


Figure: E.21: Results obtained from the present experiment on Gaussian noise, mean 0, variance 0.1

Experiment with Gaussian noise (variance 0 and mean 0), shows the real and distorted image using Gaussian noise Figure: E.19, represents Screenshot Of Edge And Corner Detection Of Image Figure: E.20, show the results obtained from the present experiment on Gaussian noise, mean 0, variance 0.1 Figure: E.21

Screenshot of original Image and with Gaussian Noise with Variance 0.5 and Mean 0:

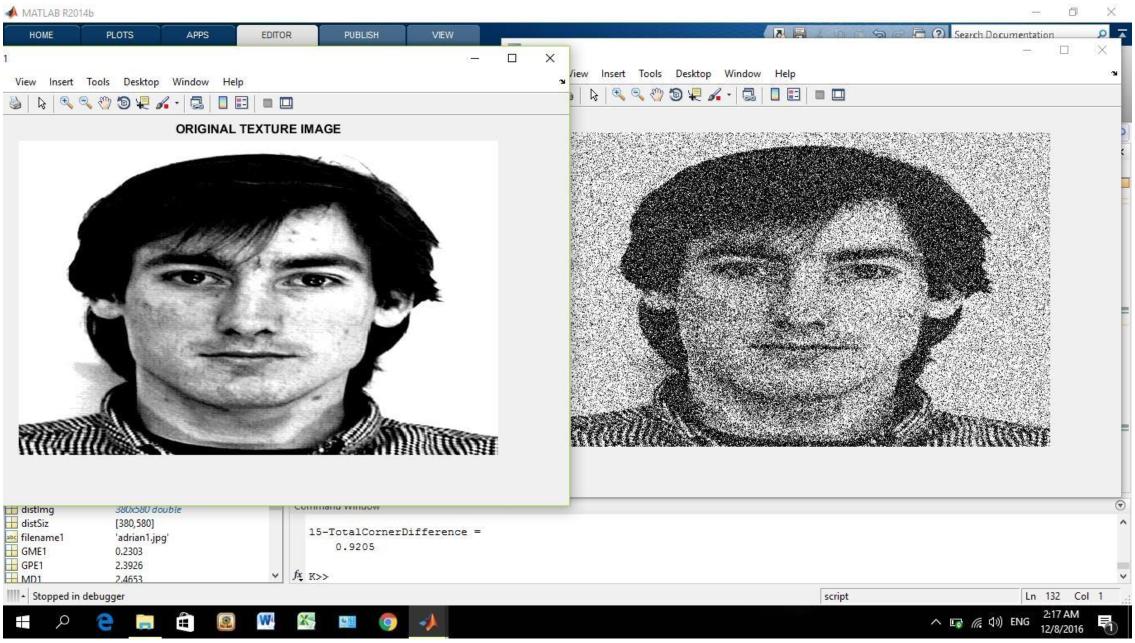


Figure: E.22: Real and distorted image using Gaussian noise

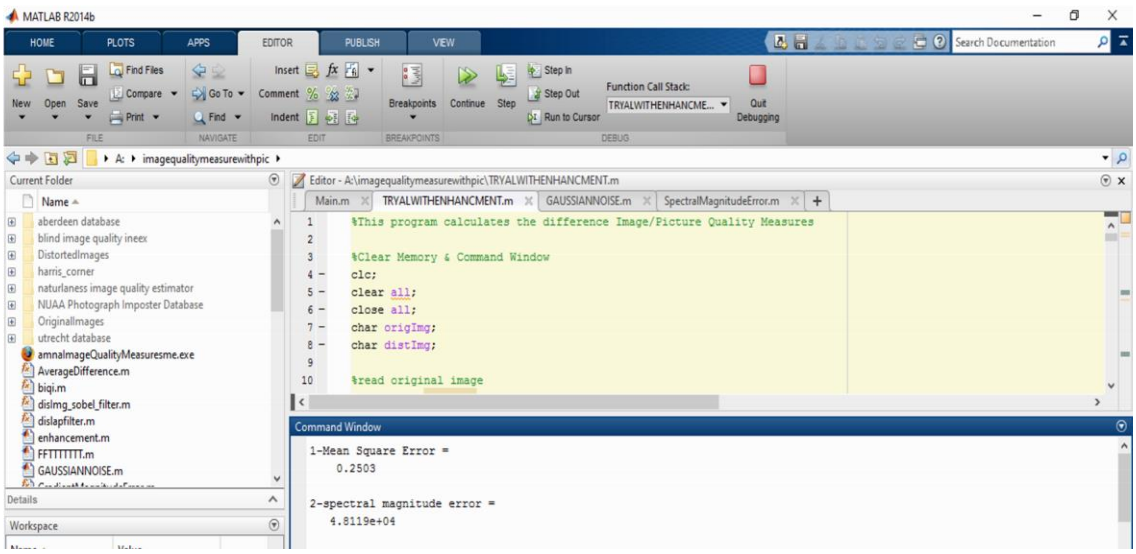


Figure: E.23: Results obtained from the present experiment on Gaussian noise, mean 0, variance 0.5

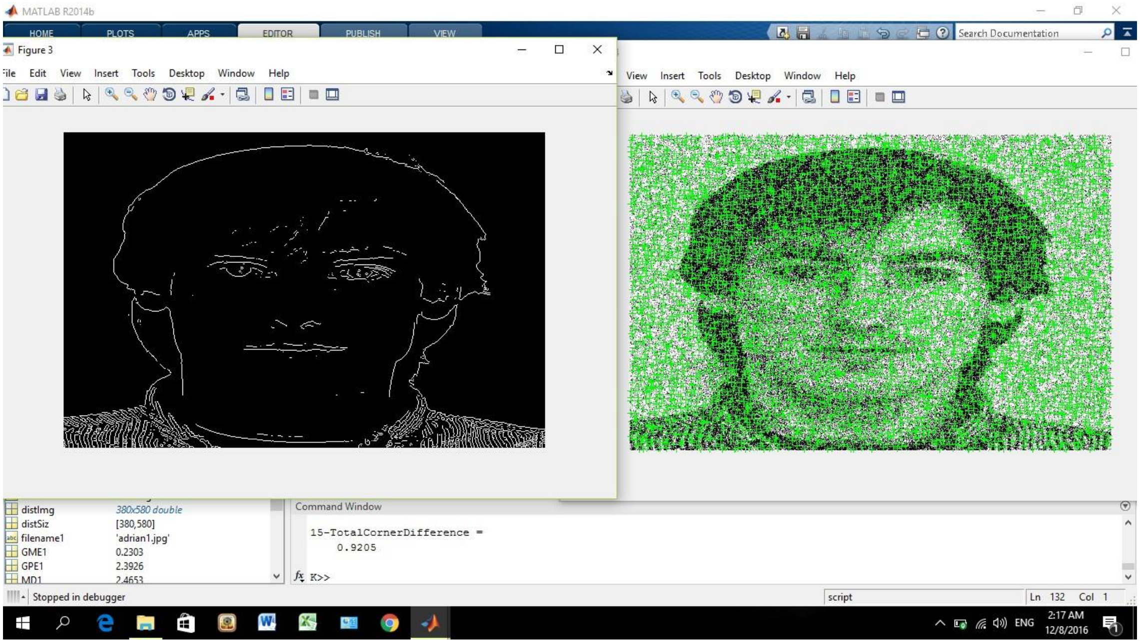


Figure: E.24: Represents screenshot of edge and corner detection of image

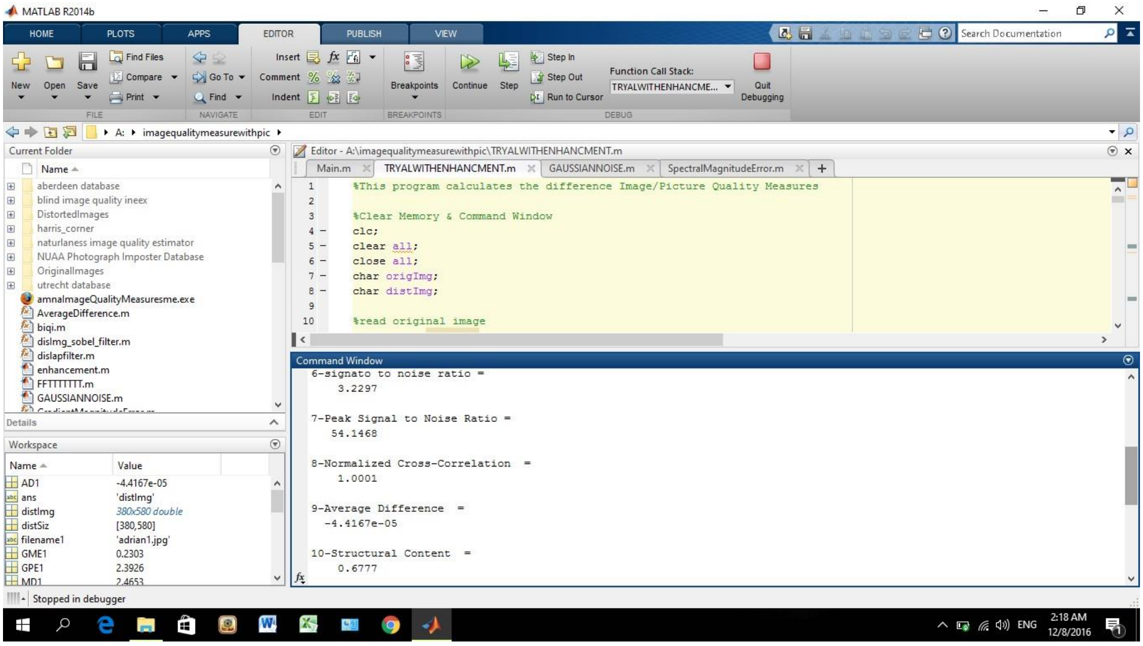


Figure: E.25: Results obtained from the present experiment on Gaussian noise, mean 0, variance 0.5

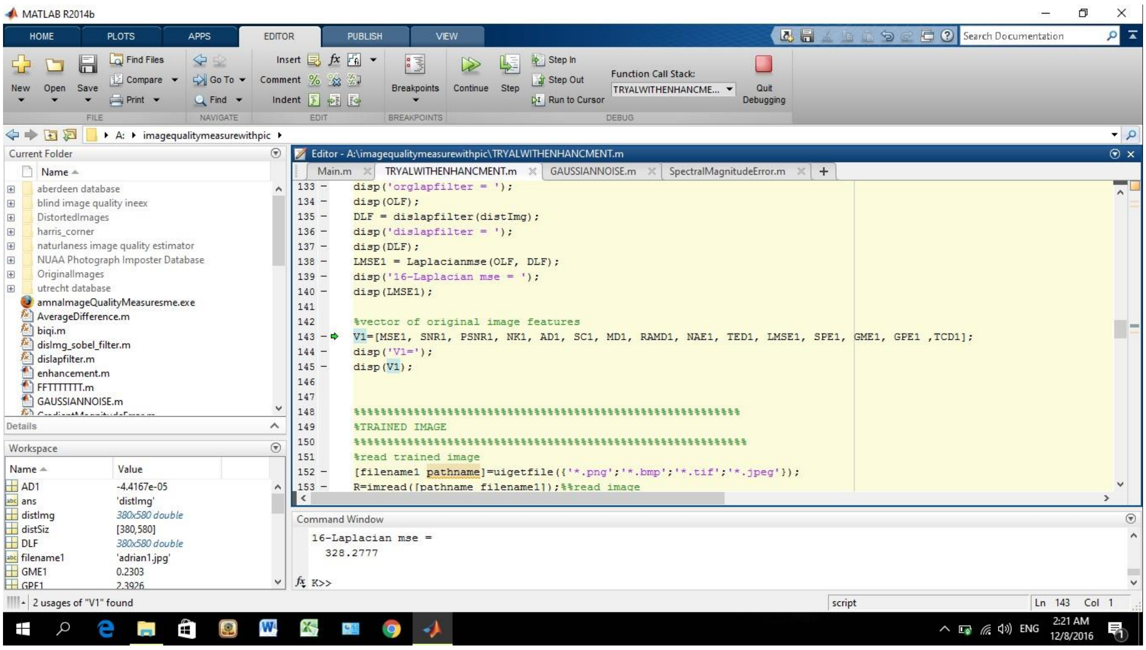


Figure: E.25: Results obtained from the present experiment on Gaussian noise, mean 0, variance 0.5

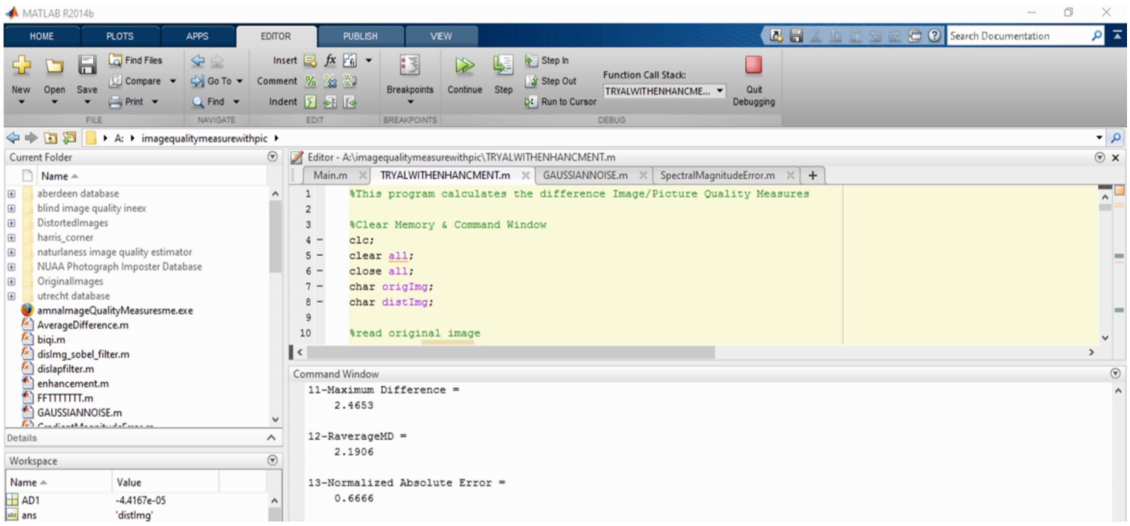


Figure: E.26: Results obtained from the present experiment on Gaussian noise, mean 0, variance 0.5

Experiment with Gaussian noise (variance 0 and mean 0), shows the real and distorted image using Gaussian noise Figure: E.22, represents Screenshot Of Edge And Corner Detection Of Image Figure: E.23, show the results obtained from the present experiment on Gaussian noise, mean 0, variance 0.5 Figure: E.24, 25, 26

Screenshot of original image and with Gaussian noise with variance 1 and mean 0:

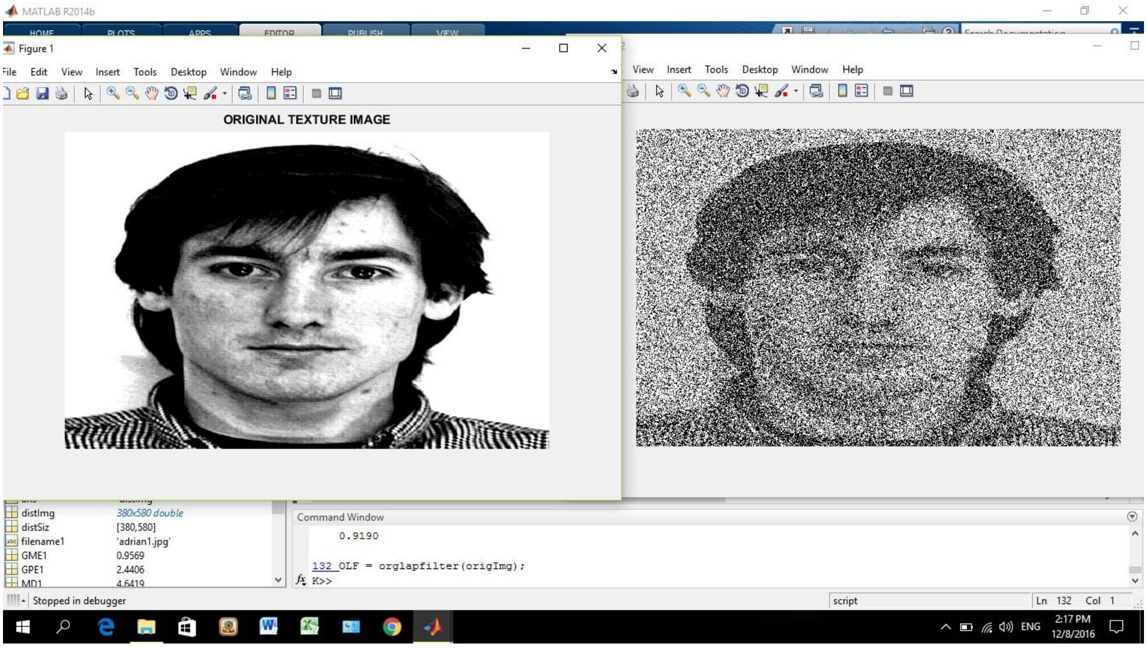


Figure: E.27 : the real and distorted image using Gaussian noise

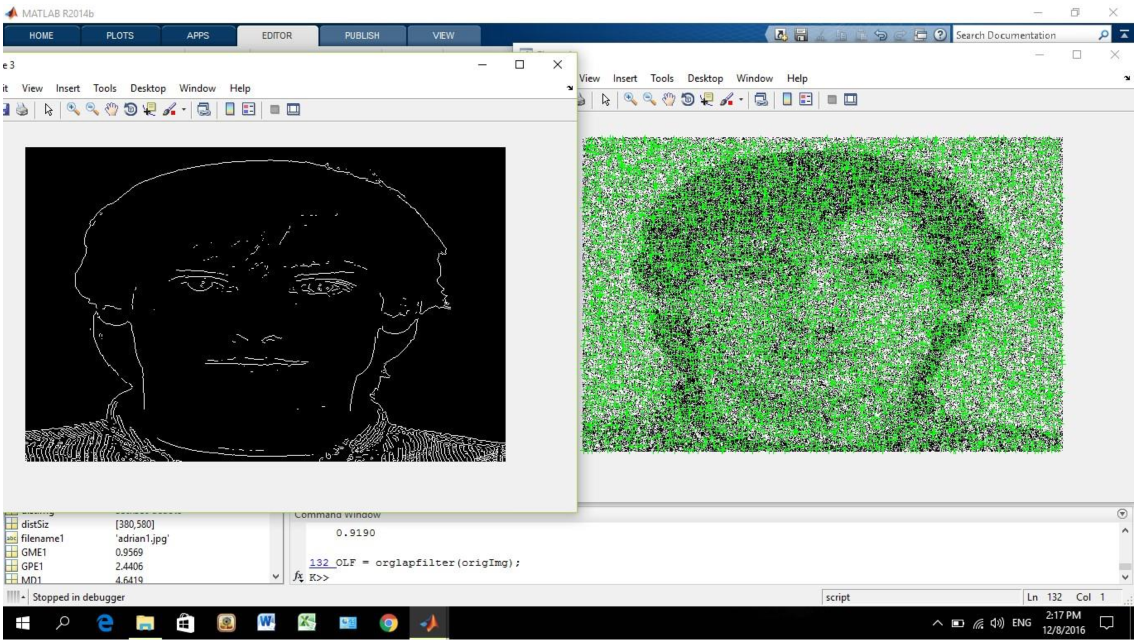


Figure: E.28: Represents screenshot of edge and corner detection of

image

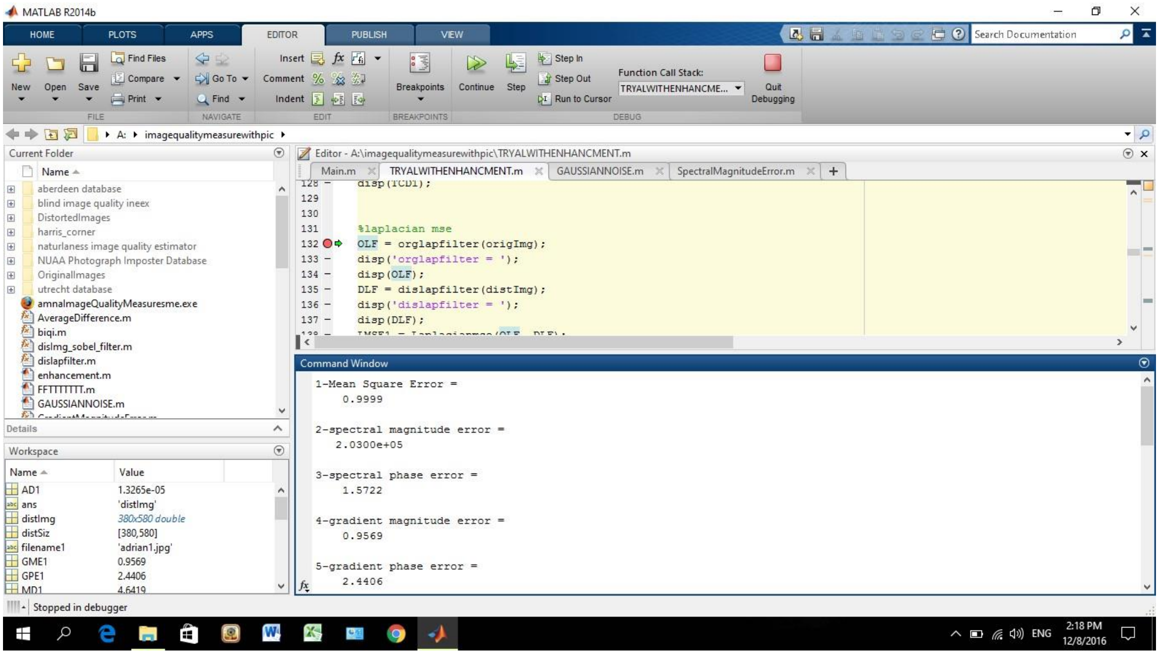


Figure: E.29: Results obtained from the present experiment on Gaussian noise, mean 0, variance 1

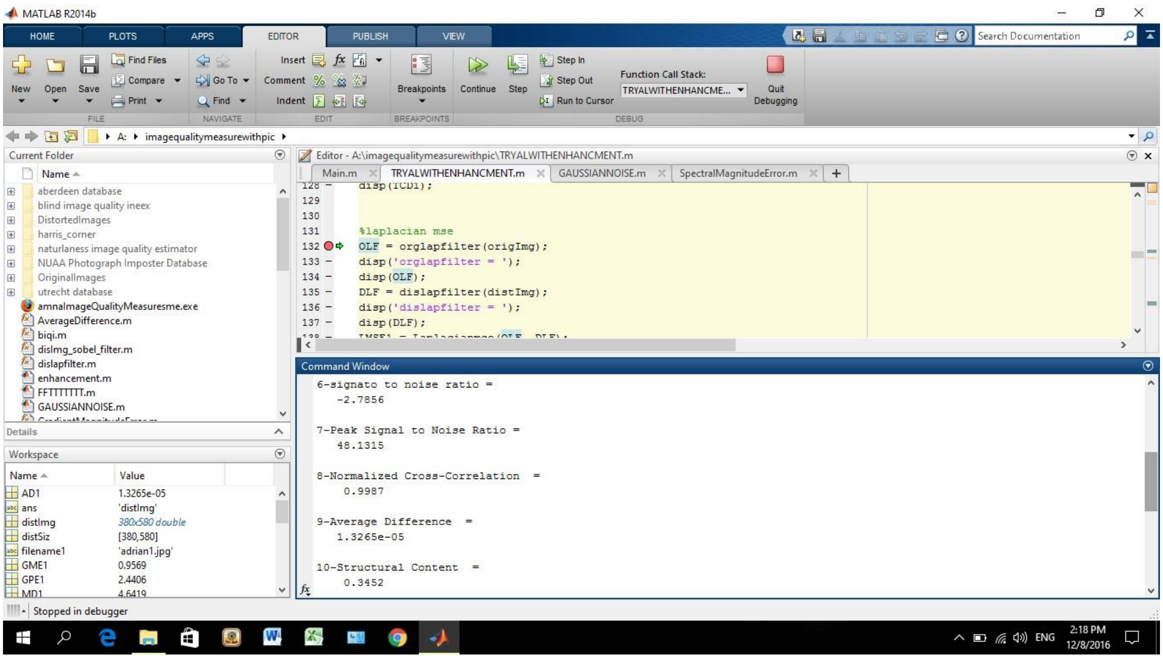


Figure: E.30: Results obtained from the present experiment on Gaussian noise, mean 0, variance 1

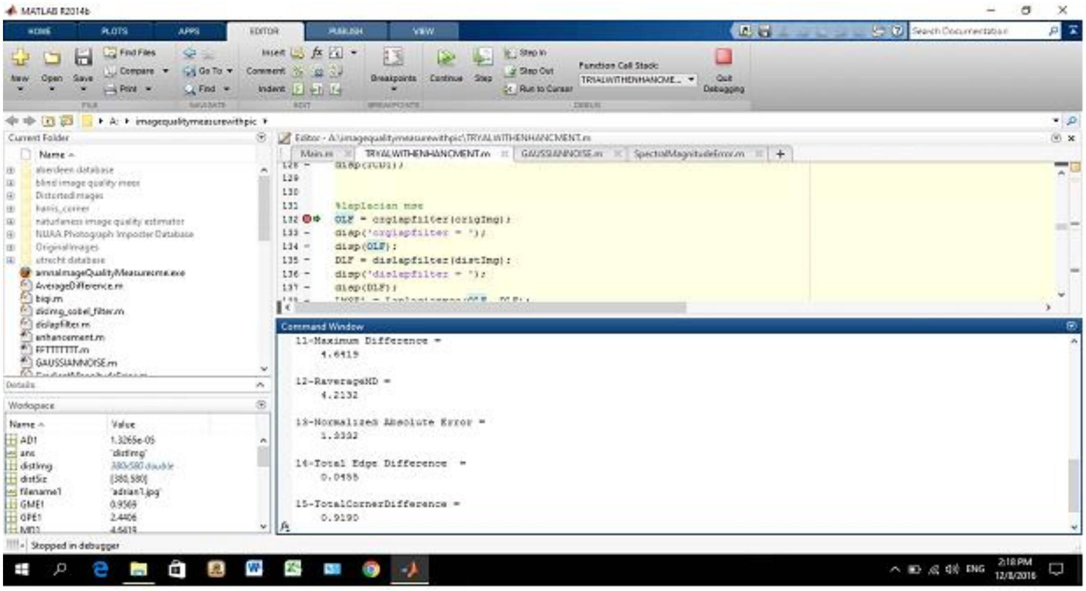


Figure: E.31: Results obtained from the present experiment on Gaussian noise, mean 0, variance 1

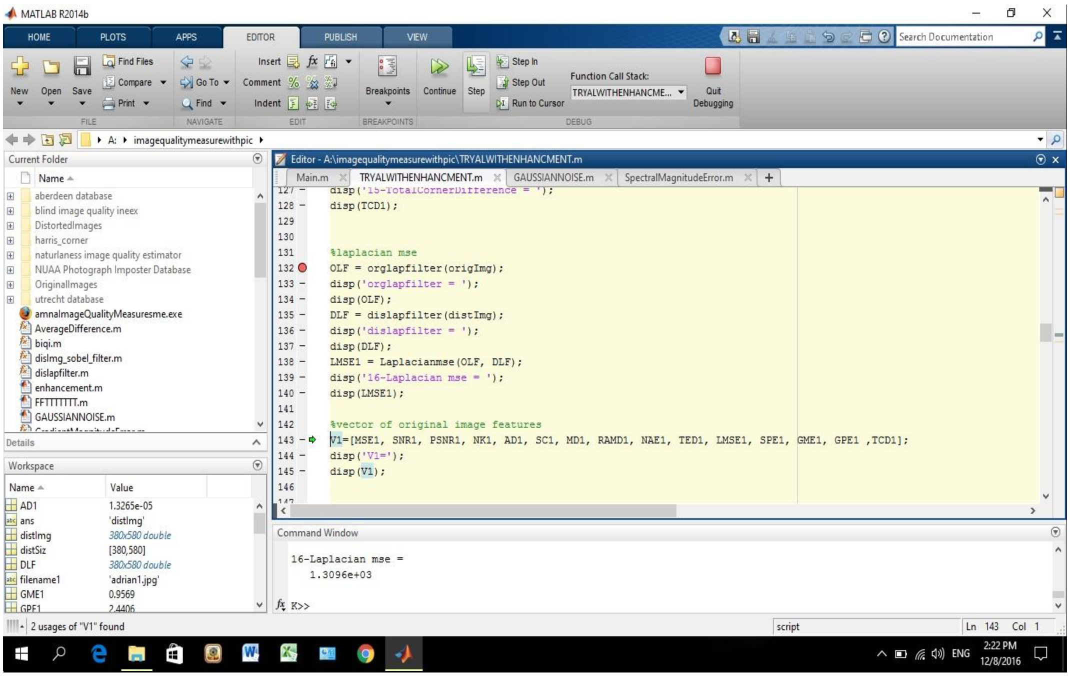


Figure: E.32: Results obtained from the present experiment on Gaussian noise, mean 0, variance 1

Experiment with Gaussian noise (variance 0 and mean 0), shows the real and distorted image using Gaussian noise Figure: E.27 , represents Screenshot Of Edge And Corner Detection Of Image Figure: E.28, show the results obtained from the present experiment on Gaussian noise, mean 0, variance 1 Figure: E.29, 30, 31,