2014

Comparison of Computer-Based and Optical Face Recognition Paradigms

Abdulaziz A. Alorf

Michigan Technological University

Copyright 2014 Abdulaziz A. Alorf

Recommended Citation


http://digitalcommons.mtu.edu/etds/743

Follow this and additional works at: http://digitalcommons.mtu.edu/etds

Part of the Electrical and Computer Engineering Commons, Optics Commons, and the Robotics Commons
COMPARISON OF COMPUTER-BASED AND OPTICAL FACE RECOGNITION PARADIGMS

By

Abdulaziz A. Alorf

A THESIS
Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
In Electrical Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY
2014

© 2014 Abdulaziz A. Alorf
This thesis has been approved in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE in Electrical Engineering.

Department of Electrical and Computer Engineering

Thesis Advisor:  Professor, Michael C. Roggemann

Committee Member:  Associate Professor, Christopher T. Middlebrook

Committee Member:  Assistant Professor, Timothy Havens

Committee Member:  Associate Professor, Aleksandr V. Sergeyev

Department Chair:  Professor, Daniel R. Fuhrmann
To my mother, father and wife.
<table>
<thead>
<tr>
<th>Contents</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>xiii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xvii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>xix</td>
</tr>
<tr>
<td>Abstract</td>
<td>xxi</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.1.1 Digital Processing</td>
<td>1</td>
</tr>
<tr>
<td>1.1.2 Optical Processing</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Problem Statement</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Technical Approach</td>
<td>3</td>
</tr>
<tr>
<td>1.3.1 Digital Model</td>
<td>3</td>
</tr>
<tr>
<td>1.3.1.1 Introduction</td>
<td>3</td>
</tr>
<tr>
<td>1.3.1.2 Image Compression</td>
<td>3</td>
</tr>
<tr>
<td>1.3.1.3 Face Recognition</td>
<td>4</td>
</tr>
<tr>
<td>1.3.1.4 Image Detection</td>
<td>4</td>
</tr>
<tr>
<td>1.3.2 Optical Model</td>
<td>5</td>
</tr>
<tr>
<td>1.3.2.1 Introduction</td>
<td>5</td>
</tr>
<tr>
<td>1.3.2.2 Face Recognition</td>
<td>5</td>
</tr>
<tr>
<td>1.3.2.3 Object Detection</td>
<td>5</td>
</tr>
<tr>
<td>1.4 Summary of Key Results</td>
<td>6</td>
</tr>
<tr>
<td>1.5 Organization</td>
<td>6</td>
</tr>
<tr>
<td>2 Theoretical Background</td>
<td>7</td>
</tr>
<tr>
<td>2.1 Preview</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Digital Processing</td>
<td>7</td>
</tr>
<tr>
<td>2.2.1 Introduction</td>
<td>7</td>
</tr>
</tbody>
</table>
2.2.2 A Principal Components Analysis (PCA) Algorithm .......................... 7
  2.2.2.1 Introduction .............................................................................. 7
  2.2.2.2 Analysis of the PCA Algorithm .............................................. 8
  2.2.2.3 An Example of the PCA Algorithm ......................................... 10
    2.2.2.3.1 Taking All Eigenvectors as Principal Components for Doing PCA ..... 14
  2.2.2.4 Reconstruction of the Data Set $D$ ........................................... 15
  2.2.2.5 Reconstruction of the Data Set $D$ of the Example in Sub-subsection 2.2.2.3 .............. 16
    2.2.2.5.1 Using All Eigenvectors for Reconstructing the Data Set $D$ ............................. 16
  2.2.2.6 Application of the PCA Algorithm to Images ............................ 17
  2.2.2.7 Reconstruction of the Original Images ...................................... 20

2.2.3 An Improved Principal Components Analysis (IPCA) Algorithm .......... 21
  2.2.3.1 Introduction .............................................................................. 21
  2.2.3.2 Analysis of the IPCA Algorithm .............................................. 21
  2.2.3.3 Applying IPCA on the Example in Sub-subsection 2.2.2.3 .............. 22

2.2.4 Comparison of the PCA and IPCA Algorithms .............................. 22

2.2.5 Applications of the PCA and IPCA Algorithms ............................... 23
  2.2.5.1 Introduction .............................................................................. 23
  2.2.5.2 Image Compression ................................................................. 23
  2.2.5.3 Face Recognition ..................................................................... 23
    2.2.5.3.1 Setting up a Recognition Threshold ....................................... 25
  2.2.5.4 Image Detection ..................................................................... 26

2.2.6 Performance Analysis of the PCA and IPCA Algorithms .................. 26
  2.2.6.1 Introduction .............................................................................. 26
  2.2.6.2 Analysis of Compression Performance ...................................... 27
    2.2.6.2.1 Introduction ..................................................................... 27
    2.2.6.2.2 Speed of Compression and Reconstruction ......................... 27
    2.2.6.2.3 Quality of a Reconstructed Image ....................................... 27
    2.2.6.2.4 Size of Compression ............................................................. 27
      2.2.6.2.4.1 Introduction ................................................................. 27
      2.2.6.2.4.2 Information Rate .......................................................... 27
      2.2.6.2.4.3 Mean Squared Error (MSE) of Compressed Images ............... 28
  2.2.6.3 Analysis of Recognition Performance ........................................ 31
    2.2.6.3.1 Introduction ..................................................................... 31
    2.2.6.3.2 Speed of Recognition .......................................................... 31
    2.2.6.3.3 Error Rate ........................................................................... 31
  2.2.6.4 Analysis of Detection Performance .......................................... 31
    2.2.6.4.1 Introduction ..................................................................... 31
    2.2.6.4.2 Speed of Detection ............................................................. 32
Contents ix

2.2.6.4.3 Error Rate ........................................ 32
2.3 Analog Optical Information Processing .......................... 32
  2.3.1 Introduction ........................................... 32
  2.3.2 Coherent and Incoherent Optical Image Processing Systems . 32
  2.3.3 Coherent Optical Information Processing Systems .......... 33
  2.3.4 The Joint Transform Correlator (JTC) .................... 35
    2.3.4.1 Sampling Issues .................................... 40
      2.3.4.1.1 Introduction .................................... 40
      2.3.4.1.2 Sampling of the Input Plane $P_1$ .................. 40
      2.3.4.1.3 Sampling of the Back Focal Plane $P_2$ ............. 41
      2.3.4.1.4 Sampling of the Back Focal Plane $P_3$ ............. 43
    2.3.4.2 Applications of the JTC ................................ 44
      2.3.4.2.1 Introduction .................................... 44
      2.3.4.2.2 Face Recognition .............................. 44
        2.3.4.2.2.1 Setting up a Recognition Threshold ............. 45
    2.3.4.3 Analysis of JTC Performance .......................... 46
      2.3.4.3.1 Introduction .................................... 46
      2.3.4.3.2 Analysis of Recognition Performance .............. 46
        2.3.4.3.2.1 Improvement of Recognition Performance .......... 46
      2.3.4.3.3 Analysis of Detection Performance ............... 47
        2.3.4.3.3.1 Improvement of Detection Performance .......... 47
  3 Modelling ................................................. 49
    3.1 Digital Modelling ...................................... 49
      3.1.1 Introduction ........................................ 49
      3.1.2 Application of the PCA Algorithm to Images .............. 49
      3.1.3 Application of the IPCA Algorithm to Images .......... 55
      3.1.4 Comparison of the PCA and IPCA Algorithms .............. 56
      3.1.5 Image Compression .................................... 58
      3.1.6 Face Recognition ..................................... 58
      3.1.7 Image Detection ....................................... 59
    3.2 Optical Modelling ...................................... 60
      3.2.1 Introduction ........................................ 60
      3.2.2 Simulation of the JTC ................................ 60
      3.2.3 Face Recognition ..................................... 65
      3.2.4 Object Detection ..................................... 67
4 Results of Performance Analysis .................................................. 69
  4.1 Performance Results of the PCA and IPCA Algorithms ................. 69
    4.1.1 Introduction ....................................................................... 69
    4.1.2 Results of Compression Performance ................................. 69
      4.1.2.1 Speed of Compression and Reconstruction ................. 69
      4.1.2.2 Quality of a Reconstructed Image ............................ 69
      4.1.2.3 Size of Compression .............................................. 72
      4.1.2.3.1 Information Rate ............................................. 72
      4.1.2.3.2 Mean Squared Error (MSE) of Compressed Images ....... 72
    4.1.3 Results of Recognition Performance .................................... 73
      4.1.3.1 Speed of Recognition .......................................... 73
      4.1.3.2 Error Rate ...................................................... 73
    4.1.4 Results of Detection Performance ........................................ 74
      4.1.4.1 Speed of Detection ............................................. 74
      4.1.4.2 Error Rate ...................................................... 75
  4.2 Results of JTC Performance .................................................... 76
    4.2.1 Results of Recognition Performance .................................... 76
      4.2.1.1 Improvement of Recognition Performance ................. 76
    4.2.2 Results of Detection Performance ........................................ 77
      4.2.2.1 Improvement of Detection Performance ....................... 77
5 Conclusion ................................................................................. 79
  5.1 Discussion of Results ............................................................. 79
    5.1.1 Introduction ....................................................................... 79
    5.1.2 Comparison of the PCA and IPCA Algorithms ....................... 79
      5.1.2.1 Introduction ...................................................... 79
      5.1.2.2 Results of Image Compression .................................... 79
      5.1.2.3 Results of Face Recognition ..................................... 80
      5.1.2.4 Results of Face Detection ....................................... 80
    5.2 Methods to Improve the Digital and the Optical Models ............. 81
      5.2.1 Introduction ................................................................... 81
      5.2.2 Improvement of the Digital and the Optical Models ........... 81
    5.3 The Digital Model Versus the Optical Model ............................. 82
    5.4 Future Work ......................................................................... 82
Bibliography ................................................................................. 83
Appendix A A Code for the Digital Model ....................................... 85
Appendix B Databases for the Digital and Optical Models ................. 156
Appendix C Results of the Digital Recognition .................................. 160
<table>
<thead>
<tr>
<th>Appendix</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>Results of the Digital Detection</td>
<td>166</td>
</tr>
<tr>
<td>E</td>
<td>Derivation of $U_2(x_2, y_2)$ for the JTC</td>
<td>172</td>
</tr>
<tr>
<td>F</td>
<td>A Code for the Optical Model</td>
<td>174</td>
</tr>
<tr>
<td>G</td>
<td>Results of the Optical Recognition</td>
<td>201</td>
</tr>
<tr>
<td>H</td>
<td>Results of the Optical Detection</td>
<td>205</td>
</tr>
<tr>
<td>I</td>
<td>MSEs Plots of Reconstructing Some Training Faces</td>
<td>209</td>
</tr>
<tr>
<td>J</td>
<td>A Code for Optimizing the Optical Model Database</td>
<td>217</td>
</tr>
<tr>
<td>K</td>
<td>Authorizations for Using People Photos</td>
<td>231</td>
</tr>
</tbody>
</table>
List of Figures

2.1 The plot of the data set \( D \). ............................................... 10
2.2 The plot of the centered data set \( R \). ................................. 11
2.3 The plot of the centered data set \( R \) as well as the orthonormal eigenvectors. ................................................. 13
2.4 The projected centered data by using just the highest correlated eigenvector \( v_1 \). .......................................................... 14
2.5 The projected centered data when all eigenvectors are used as principal components. ........................................... 15
2.6 The reconstructed data by using just the highest correlated eigenvector \( v_1 \). .......................................................... 17
2.7 A normalized training face and its normalized corresponding unknown face image. ................................................. 24
2.8 An unnormalized training face and its unnormalized corresponding unknown face image. ............................................ 25
2.9 The architecture for coherent optical information processing. ................................................................. 34
2.10 The joint transform correlator: (a) Recording the filter; and (b) getting the filter output. ........................................... 36
2.11 The bandwidths of the patterns of the cross-correlated field in the output plane \( P_3 \) along the spatial coordinates \( x_3 \) and \( y_3 \). .......... 39
2.12 The alignment of the input transparencies in the input plane \( P_1 \) . . . 41
2.13 The alignment of the patterns of the cross-correlated field in the output plane \( P_3 \). .......................................................... 42
3.1 The database of the training faces. ................................. 50
3.2 The average histogram for all training faces. ......................... 50
3.3 The application of the threshold on the training faces. .............. 51
3.4 The normalization of all training faces by means of the selected threshold. ...................................................... 52
3.5 The centered training faces. ............................................. 53
3.6 The average training face. Note that, this is a negative image. ...... 53
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.7</td>
<td>The centered (with respect to the set of the training faces) training faces. Note that, these are negative images.</td>
</tr>
<tr>
<td>3.8</td>
<td>The eigenfaces. Note that, these are negative images.</td>
</tr>
<tr>
<td>3.9</td>
<td>A comparison between the biggest calculated eigenvalues by using the PCA and IPCA algorithms as well as the calculated eigenvalues from the covariance matrix for all training faces.</td>
</tr>
<tr>
<td>3.10</td>
<td>Some samples of the tested images.</td>
</tr>
<tr>
<td>3.11</td>
<td>A focused lens.</td>
</tr>
<tr>
<td>3.12</td>
<td>The desired impulse response $h$.</td>
</tr>
<tr>
<td>3.13</td>
<td>The object $g$.</td>
</tr>
<tr>
<td>3.14</td>
<td>The transmitted field $U_1(x_1, y_1)$ from the input plane $P_1$.</td>
</tr>
<tr>
<td>3.15</td>
<td>The incident intensity $I(x_2, y_2)$ on the back focal plane $P_2$.</td>
</tr>
<tr>
<td>3.16</td>
<td>The cross-correlated field $U_3(x_3, y_3)$ in the back focal plane $P_3$.</td>
</tr>
<tr>
<td>3.17</td>
<td>The adaptive filtering mask.</td>
</tr>
<tr>
<td>3.18</td>
<td>The filtered cross-correlated field in the back focal plane $P_3$.</td>
</tr>
<tr>
<td>3.19</td>
<td>The database of the impulses.</td>
</tr>
<tr>
<td>3.20</td>
<td>Some samples of the objects.</td>
</tr>
<tr>
<td>4.1</td>
<td>The plot of the mean squared errors (MSEs) of reconstructing training face number five for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used.</td>
</tr>
<tr>
<td>4.2</td>
<td>The reconstruction of training face number five by using the highest correlated eigenface ($q = 1$), the PCA eigenfaces ($q = 8$), the IPCA eigenfaces ($q = 7$), and all eigenvectors as eigenfaces ($q = 2500$); along with mean squared errors resulted from reconstructing the training face by using those eigenfaces.</td>
</tr>
<tr>
<td>4.3</td>
<td>The rates of information for different selected eigenfaces compared with resulted information rates when the PCA and IPCA algorithms are used.</td>
</tr>
<tr>
<td>4.4</td>
<td>The mean squared errors (MSEs) of compression for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used.</td>
</tr>
<tr>
<td>4.5</td>
<td>The error rates of recognition for different selected eigenfaces compared with resulted error rates when the PCA and IPCA algorithms are used.</td>
</tr>
<tr>
<td>4.6</td>
<td>The error rates of detection for different selected eigenfaces compared with resulted error rates when the PCA and IPCA algorithms are used.</td>
</tr>
<tr>
<td>4.7</td>
<td>The database of the optimal impulses for face recognition.</td>
</tr>
<tr>
<td>4.8</td>
<td>The database of the optimal impulses for object detection.</td>
</tr>
<tr>
<td>B.1</td>
<td>Images for Mr. Mansour Alshammari, 1st projection.</td>
</tr>
<tr>
<td>B.2</td>
<td>Images for Mr. Mansour Alshammari, 2nd projection.</td>
</tr>
<tr>
<td>B.3</td>
<td>Images for Mr. Mansour Alshammari, 3rd projection.</td>
</tr>
</tbody>
</table>
B.4 Images for Mr. Methkir Alharthee, 1\textsuperscript{st} projection. ........... 157
B.5 Images for Mr. Methkir Alharthee, 2\textsuperscript{nd} projection. ........... 158
B.6 Images for Mr. Methkir Alharthee, 3\textsuperscript{rd} projection. ........... 158
B.7 Images for Mr. Mohammed Hanafy, 1\textsuperscript{st} projection. ........... 158
B.8 Images for Mr. Mohammed Hanafy, 2\textsuperscript{nd} projection. .......... 159
B.9 Images for Mr. Mohammed Hanafy, 3\textsuperscript{rd} projection. ........... 159

I.1 The plot of the mean squared errors (MSEs) of reconstructing training face number one for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used. ..................................... 209
I.2 The plot of the mean squared errors (MSEs) of reconstructing training face number two for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used. .......................................................... 210
I.3 The plot of the mean squared errors (MSEs) of reconstructing training face number three for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used. .......................................................... 211
I.4 The plot of the mean squared errors (MSEs) of reconstructing training face number four for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used. .......................................................... 212
I.5 The plot of the mean squared errors (MSEs) of reconstructing training face number six for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used. .......................................................... 213
I.6 The plot of the mean squared errors (MSEs) of reconstructing training face number seven for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used. .......................................................... 214
I.7 The plot of the mean squared errors (MSEs) of reconstructing training face number eight for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used. .......................................................... 215
I.8 The plot of the mean squared errors (MSEs) of reconstructing training face number nine for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used. .......................................................... 216

K.1 The authorization for using the photos of Mr. Mansour Thuwaini Al-Shammari. .......................................................... 231
K.2 The authorization for using the photos of Mr. Mathkar Alawi Alharthi. 232
K.3 The authorization for using the photos of Mr. Mohamed Elsayed Hanafy.
List of Tables

2.1 The calculations of the $VCR$ and $TVC$. ........................................ 12
2.2 The calculations of the $IR$ and $AIR$. ........................................ 22

3.1 The calculations of the $VCR$ and $TVC$. ........................................ 55
3.2 The calculations of the $IR$ and $AIR$. ........................................ 57
3.3 The recognition of the tested images in Figure 3.10. ......................... 58
3.4 The detection of the tested images in Figure 3.10. ......................... 60
3.5 The recognition of the objects in Figure 3.20. ................................ 66
3.6 The detection of the objects in Figure 3.20. ................................ 67

5.1 The results of image compression. .................................................. 80
5.2 The results of face recognition. ...................................................... 80
5.3 The results of face detection. ......................................................... 81
5.4 The comparison between the digital and the optical models. ............ 82

C.1 The recognition of all 180 tested images by using the PCA algorithm. 160
D.1 The detection of all 180 tested images by using the PCA algorithm. . 166

G.1 The recognition of all 108 objects by using the joint transform correlator (JTC). ......................................................... 201
H.1 The detection of all 108 objects by using the joint transform correlator (JTC). ......................................................... 205
Acknowledgements

There are a number of people without whom this thesis might not have been written and to whom I am greatly indebted. I would like to take this opportunity to look back and acknowledge people who have helped to shape me into the successful young man I am today.

First of all, I am grateful to the Almighty God for all the blessings that you have bestowed on me.

A very special thank you to my parents who continue to grow, learn and develop; and who have been a source of encouragement and inspiration to me throughout my life.

To my dear wife, Hajar, who remains willing to engage with the struggle and to tolerate my long busyness in research. Thank you so much for your practical and emotional support.

A special note of thanks to my sponsor, Qassim University in Saudi Arabia for granting me this scholarship to pursue my graduate studies. Thank you from the bottom of my heart for supporting me since I was one of your students at the College of Engineering.

I would like to express my deepest appreciation to Prof. Sulaiman A. Al-Yahya, the vice president of Qassim University for planning, development and quality; and Dr. Abdulrahman F. Almarshoud, the dean of Engineering College at Qassim University; for their continuous support. You are a source of encouragement and inspiration for me. Everytime I visit or call, you try your best to push me forward regardless of any difficulty.

My journey here at Michigan Technological University has been challenging. I have enjoyed all of the opportunities that have come my way. I have acquired such an immense wealth of knowledge and experience throughout my time here that I will likely owe most of my future professional success to the four years I have spent up here in the freezing cold. I want to take this opportunity to send my warmest thanks to this institution.

I wish to express my sincere thanks to my advisor Prof. Michael C. Roggemann
for his kindness, expertise, sincerity, valuable guidance and encouragement. You have actively supported me in my determination to find and realize my potential and to make this contribution to our world. I recall that when I faced a problem in my research, I would email you with a description of the problem; then you sometimes replied with “Let’s chat about that tomorrow in my office” which means that it is not an easy problem. I want to tell you that I really hate the word “chat” because it means that I have to afford more effort and time. I am really proud that you have directed my thesis.

I would like to say thank you to my friends Mr. Mohamed Elsayed Hanafy, Mr. Mathkar Alawi Alharthi and Mr. Mansour Thuwaini Al-Shammar for providing me with images of their faces to use in my research. It is impossible to forget that when I call you to schedule a session for taking some photos, you were willing to come and offer any time I needed. I really appreciate your cooperation.

I would like to thank Dr. Jeffrey B. Burl for providing me with essential suggestions on different aspects of my research.

I also want to thank Dr. Glen E. Archer for teaching me some powerful research tools. Thank you very much for offering me time and lots of effort.

Thanks with all of my heart to Dr. Imran Aslam and Mr. Ehsan Taheri for indispensable advice, information and support throughout my master’s degree.

I take this opportunity to record my sincere thanks to my committee members Dr. Christopher T. Middlebrook, Dr. Timothy Havens and Dr. Aleksandr V. Sergeyev for their willingness to be on the committee and taking the time to work on my thesis.

I am also very grateful to Mr. Jim LaBeske and Ms. Jamie Peryam who knowingly and unknowingly have made my and my family’s stay in Houghton a lot easier than we thought it was going to be.

Finally, I also place on record my sense of gratitude to one or all who directly or indirectly have lent their helping hand in this thesis or have supported me.
Abstract

The main objectives of this thesis are to validate an improved principal components analysis (IPCA) algorithm on images; designing and simulating a digital model for image compression, face recognition and image detection by using a principal components analysis (PCA) algorithm and the IPCA algorithm; designing and simulating an optical model for face recognition and object detection by using the joint transform correlator (JTC); establishing detection and recognition thresholds for each model; comparing between the performance of the PCA algorithm and the performance of the IPCA algorithm in compression, recognition and, detection; and comparing between the performance of the digital model and the performance of the optical model in recognition and detection. The MATLAB © software was used for simulating the models.

PCA is a technique used for identifying patterns in data and representing the data in order to highlight any similarities or differences. The identification of patterns in data of high dimensions (more than three dimensions) is too difficult because the graphical representation of data is impossible. Therefore, PCA is a powerful method for analyzing data. IPCA is another statistical tool for identifying patterns in data. It uses the information theory for improving PCA. The joint transform correlator (JTC) is an optical correlator used for synthesizing a frequency plane filter for coherent optical systems.

The IPCA algorithm, in general, behaves better than the PCA algorithm in the most of the applications. It is better than the PCA algorithm in image compression because it obtains higher compression, more accurate reconstruction, and faster processing speed with acceptable errors; in addition, it is better than the PCA algorithm in real-time image detection due to the fact that it achieves the smallest error rate as well as remarkable speed. On the other hand, the PCA algorithm performs better than the IPCA algorithm in face recognition because it offers an acceptable error rate, easy calculation, and a reasonable speed. Finally, in detection and recognition, the performance of the digital model is better than the performance of the optical model.
Chapter 1. Introduction

1.1 Background

1.1.1 Digital Processing

Principal components analysis (PCA) [1] and improved principal components analysis (IPCA) [2] are statistical tools frequently used for analyzing data. Their main applications are pattern recognition such as face detection and recognition, and data compression such as image compression.

PCA is a technique used for identifying patterns in data and representing the data in such a way that their similarities and differences are highlighted. The identification of patterns in data of high dimensions (more than three dimensions) is too difficult because the graphical representation of data is impossible. Therefore, PCA is a powerful method for analyzing data. The PCA algorithm starts with the creation of a data set and ends with the projection of the data on the eigenspace. A covariance matrix is computed for the data; in addition, the eigenvectors and eigenvalues of the covariance matrix are obtained. Eigenvectors associated with the biggest eigenvalues of the covariance matrix follow the most significant patterns of the data. Those eigenvectors are called the principle components of the data set. Therefore, the eigenvalues of the covariance matrix work as measures of how much information is contained in each of the principal components. The principal components form a feature vector matrix. In order to select principal components that form the feature vector matrix, the variance contribution rate ($VCR$) and the total variance contribution rate ($TVC$) (they are proposed in the IEEE paper presented in Reference [2]) are computed. When the $TVC$ is significantly high then $q$ eigenvectors associated with the biggest $q$ eigenvalues can be selected. The feature vector matrix is used for projecting the data on the eigenspace. Finally, by projecting the data on the eigenspace, the PCA algorithm is completed.
Section 1.2. Problem Statement

IPCA is another statistical tool for identifying patterns in data. It is similar to PCA except for the way that it selects eigenvectors that form the feature vector matrix. It affords a new accurate method to measure the information content of the principal components based on the information theory for improving PCA. For measuring the degree of information content of the eigenvectors, two new concepts are used; the first is the information rate ($IR$) and the second is the accumulated information rate ($AIR$) (they are proposed in the IEEE paper presented in Reference [2]). When the $AIR$ is significantly high then $q$ eigenvectors associated with the biggest $q$ eigenvalues can be selected.

1.1.2 Optical Processing

Spatially coherent light is going to be used in the optical model. Coherent optical systems are linear in complex amplitude; therefore, filtering processes can be performed by direct manipulation of complex amplitude appearing in the back focal plane of a Fourier transforming lens. There are at least two methods for synthesizing the frequency plane filter for coherent optical systems. One of these methods is by using the joint transform correlator (JTC), Reference [3], Section 8.5.

The JTC is an optical correlator used for synthesizing the frequency plane filter for coherent optical systems. This correlator was invented by Weaver and Goodman [4]. The filter is divided into two stages: recording the filter, and getting the filter output. The transparencies of the desired impulse response $h$ and the data $g$ (here it is called the object) to be filtered are aligned simultaneously in the input plane. They are then Fourier transformed together. At that point, a spatial light modulator (SLM) captures the intensity distribution of the transformed field. The intensity is then Fourier transformed again for producing the cross-correlated field in the output plane. The output field is composed of four terms; two terms respectively represent the cross-correlation of the impulse response $h$ and itself as well as the cross-correlation of the data $g$ and itself; the third and fourth terms represent the cross-correlations of $h$ and $g$. Lastly, the joint transform correlator has a great feature: its ability to change the filter impulse response quickly. Therefore, it is considered beneficial for real-time systems. On the other hand, its defect is that the input bandwidth of the data is reduced due to the filter impulse response being introduced simultaneously with the data to be filtered.

1.2 Problem Statement

The main objectives of this thesis are to validate the improved principal components analysis (IPCA) algorithm on images; designing and simulating a digital model for image compression, face recognition, and image detection by using the principal components analysis (PCA) algorithm and the IPCA algorithm; designing and simulating
an optical model for face recognition and object detection by using the joint transform
correlator (JTC); establishing detection and recognition thresholds for each model;
comparing between the performance of the PCA algorithm and the performance of
the IPCA algorithm in compression, recognition and detection; and comparing be-
tween the performance of the digital model and the performance of the optical model
in recognition and detection.

1.3 Technical Approach

1.3.1 Digital Model

1.3.1.1 Introduction

This subsection provides a general overview of technical approaches behind the ap-
plication of the PCA and IPCA algorithms in image compression, face recognition,
and image detection.

Here, the database for each algorithm is composed of some images of faces (training
faces). The principal components that form the feature vector matrix are here called
eigenfaces.

1.3.1.2 Image Compression

When some of the eigenvectors that are calculated from the covariance matrix for all
training faces are selected to form the feature vector matrix then the dimensions of
the reconstructed data set will be reduced. This implies that the PCA and IPCA
algorithms work as compression. The algorithms are said to be lossy because a
decompressed image is not exactly the same as the original one, but is generally
worse.

Compression performance for each algorithm as analyzed from three points of view
are the speed of compression and reconstruction, the quality of a reconstructed image,
and the size of compression. The number of the eigenfaces that is used to compress
and reconstruct the training faces mainly controls the processing speed of compression
and reconstruction. When a small number of the eigenfaces is used to project and
reconstruct the training faces then the processing speed will increase and vice versa.
For measuring the quality of a reconstructed image, the mean squared error (MSE)
between the image and its reconstruction can be computed. The size of compression
can be measured in two ways: these are through the information rate and the mean
squared error (MSE) of compressed images. The information rate measures how much
information is present after compression compared with information present before
compression; in other words, it measures the number of pixels after compression
compared to before compression.
1.3.1.3 Face Recognition

The PCA and IPCA algorithms are used to recognize an unknown face image based on the database that contains the training faces. For doing face recognition, an unknown face image is taken. The training faces and the unknown face image are projected on the eigenspace by using the PCA or IPCA feature vector matrix. The Euclidean distance between the projected unknown face image and each projected training face is computed. Then the unknown face image is recognized as a training face, which has the minimum distance from the unknown face image.

Unfortunately, when the unknown face image does not have a similar training face then getting the minimum distance does not always mean that the unknown face image is recognized as a training face that has the minimum distance from the unknown face image. Therefore, a certain threshold must be used to increase the accuracy of recognition. For setting up a recognition threshold, the mean and standard deviation (the average distance from the mean to a point) are established for each training face. Then recognition can be updated as, when the obtained minimum distance between the unknown face image and a training face is less than or equal to the mean plus the standard deviation for the training face and bigger than or equal to the mean minus the standard deviation for the training face. At that point, the unknown face image is recognized as that training face; otherwise, it is an unknown face image.

Recognition performance can be analyzed from two points of view: these are the speed of recognition and the error rate. The number of selected eigenfaces that are used to recognize the unknown face image mainly controls the recognition speed. When the number of the selected eigenfaces decreases, the processing speed increases and vice versa. The error rate computes the percentage of error in recognition.

1.3.1.4 Image Detection

The PCA and IPCA algorithms are used to detect whether or not an unknown image contains a face based on a determined threshold for detection. Hence, in image detection, only a detection threshold is needed.

To obtain the detection threshold, the mean and the standard deviation are established for some images that contain faces. In regards to detection, an unknown image is taken. It is projected on the eigenspace and reconstructed again by using the PCA or IPCA feature vector matrix. Then, detection can be performed as if the Euclidean distance between the unknown image and its reconstruction is less than or equal to the computed mean plus standard deviation and bigger than or equal to the computed mean minus standard deviation then the unknown image is detected as a face image; otherwise, it is not a face image.

Detection performance can be analyzed from two points of view: these are the speed of detection and the error rate. The number of selected eigenfaces that are used to detect the unknown image mainly controls the detection speed. When the
number of selected eigenfaces decreases, the processing speed increases and vice versa. The error rate computes the percentage of error in detection.

1.3.2 Optical Model

1.3.2.1 Introduction

This subsection provides a general overview of technical approaches behind the application of the joint transform correlator (JTC) in face recognition and object detection.

1.3.2.2 Face Recognition

The joint transform correlator (JTC) is used to recognize an unknown face object based on a database of desired impulses. The database is composed of some images of faces (impulses). For face recognition, an unknown face object is picked. The cross-correlated field between the unknown face object and each impulse is obtained. Then, the unknown face object is recognized as an impulse, which has the biggest cross-correlation with the unknown face object among other impulses.

Unfortunately, when the unknown face object does not have a similar impulse response, getting the biggest cross-correlation does not always mean that the unknown face object is recognized as an impulse, which has the biggest cross-correlation with the unknown face object. Therefore, a certain threshold must be used to increase the accuracy of recognition. For setting up a recognition threshold, the mean and standard deviation (the average distance from the mean to a point) are established for each impulse. Then, recognition can be updated, when the biggest cross-correlation with an impulse is less than or equal to the mean plus the standard deviation for the impulse and bigger than or equal to the mean minus the standard deviation for the impulse then the unknown face object is recognized as that impulse response; otherwise, it is an unknown face object.

Recognition performance can be analyzed by calculating an error rate of recognition. The error rate computes the percentage of error in recognition.

1.3.2.3 Object Detection

The joint transform correlator (JTC) is used to detect whether or not an unknown object contains a face based on a determined threshold for detection. Hence, in object detection, only a detection threshold is needed.

To obtain the detection threshold, the mean and the standard deviation are established for some objects that contain faces. For doing detection, an unknown object is taken. The unknown object is cross-correlated with any impulse response. Then, detection can be performed as if the resulted cross-correlation is less than or equal to the computed mean plus standard deviation and bigger than or equal to the com-
puted mean minus standard deviation then the unknown object is detected as a face
object; otherwise, it is not a face object.

Detection performance can be analyzed by calculating an error rate of detection.
The error rate computes the percentage of error in detection.

1.4 Summary of Key Results

The IPCA algorithm, in general, behaves better than the PCA algorithm in the most
of the applications. It is better than the PCA algorithm in image compression because
it obtains higher compression, more accurate reconstruction, and faster processing
speed with acceptable errors; in addition, it is better than the PCA algorithm in
real-time image detection due to the fact that it achieves the smallest error rate as
well as remarkable speed. On the other hand, the PCA algorithm performs better
than the IPCA algorithm in face recognition because it offers an acceptable error rate,
easy calculation, and a reasonable speed. Finally, in detection and recognition, the
performance of the digital model is better than the performance of the optical model.

1.5 Organization

The remainder of this thesis is organized as follows:

- Chapter 1: provides a general overview of this thesis.
- Chapter 2: covers theoretical backgrounds behind the PCA and IPCA algo-
rithms, their applications, and their performance in the applications. A com-
parison between the PCA and IPCA algorithms is also provided in this chapter.
Finally, it shows a theoretical background for designing an optical model for
object detection and face recognition; and theories behind the joint transform
correlator (JTC), its applications, and its performance in the applications.
- Chapter 3: presents the simulations of the PCA and IPCA algorithms by us-
ing the MATLAB © software and comparison between the simulations. It also
presents the simulations of the PCA and IPCA applications by using the MAT-
LAB © software. Lastly, this chapter provides the simulations of the joint
transform correlator (JTC) and its applications by using the MATLAB © soft-
ware.
- Chapter 4: covers the performance results of the PCA and IPCA algorithms in
their applications. Also, this chapter provides the results of JTC performance
in its applications.
- Chapter 5: presents a conclusion of this thesis.
Chapter 2. Theoretical Background

Theoretical Background

2.1 Preview

Digital and optical image processing are areas used experimentally to establish solutions to given problems. In this chapter, theoretical backgrounds for a couple of digital and optical processing techniques are demonstrated.

2.2 Digital Processing

2.2.1 Introduction

Principal components analysis (PCA) and improved principal components analysis (IPCA) are statistical tools frequently used for analyzing data. Their main applications are pattern recognition such as face detection and recognition, and data compression such as image compression. It is found that IPCA acts better than PCA in the most of applications. The analysis of each one is covered in this section.

2.2.2 A Principal Components Analysis (PCA) Algorithm

2.2.2.1 Introduction

PCA is a technique used for identifying patterns in data and representing the data in such a way as to highlight their similarities and differences. The identification of patterns in data of high dimensions (more than three dimensions) is too difficult because the graphical representation of data is impossible. Therefore, PCA is a powerful method for analyzing data.

This subsection covers the steps that are needed for performing PCA on a set of data and reconstructing the data set along with examples; as well as the steps that
are needed for performing PCA on images and reconstructing the images back. How and why the technique works are explained as well as what is happening at each step is demonstrated.

### 2.2.2.2 Analysis of the PCA Algorithm

The PCA algorithm is built up based on the following steps:

1. **Step 1:** getting some data.
2. **Step 2:** computing the mean vector $\mathbf{m}_D$ of the data set as in Equation 2.1. Where $\mathbf{D}_k$ is a column vector contains one data item such that $\mathbf{D}_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix}$; and $n$ is the total number of the data items.

   $$\mathbf{m}_D = \frac{1}{n} \sum_{k=1}^{n} \mathbf{D}_k$$  \hspace{1cm} (2.1)

3. **Step 3:** subtracting the mean from each of the data dimensions as in Equation 2.2. This produces a data set whose mean is zero which means the data set is centered. This step is really an important step for decreasing the error rate of face recognition.

   $$\mathbf{R} = [\mathbf{D}_1 - \mathbf{m}_D, \ldots, \mathbf{D}_n - \mathbf{m}_D]$$  \hspace{1cm} (2.2)

4. **Step 4:** calculating a covariance matrix as in Equation 2.3. The covariance matrix is real and symmetric. $\frac{1}{n-1}$ can be removed or left because it is just a normalization factor which affects all values by the same amount. The division on $n-1$ and not $n$ because the data set is a sample of the population. It is found that gives an answer is very close to the answer that will result if the entire population is used. If the covariance matrix is calculated for the entire population then the division must be on $n$.

   $$\mathbf{C} = \frac{1}{n-1} \sum_{k=1}^{n} (\mathbf{D}_k - \mathbf{m}_D)(\mathbf{D}_k - \mathbf{m}_D)^T = \frac{1}{n-1} \times \mathbf{R}^T \mathbf{R} \hspace{1cm} (2.3)$$

5. **Step 5:** obtaining the eigenvectors and eigenvalues of the covariance matrix $\mathbf{C}$ as in Equation 2.4 and Equation 2.5 respectively. Where the columns of the matrix $\mathbf{A}$ are the eigenvectors of $\mathbf{C}$; the diagonal of the matrix $\mathbf{\Lambda}$ contains the eigenvalues of $\mathbf{C}$; and $d$ is the number of the dimensions of the data set.

   $$\mathbf{A} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \ldots, \mathbf{v}_d]$$  \hspace{1cm} (2.4)

   $$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_d \end{bmatrix}, \quad \text{where} \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$$  \hspace{1cm} (2.5)
Since the covariance matrix $C$ is a real and symmetric (Reference [5]; Pages 207 and 208; Theorems 5.9, 5.10, 5.11 and 5.12) $d \times d$ matrix then its eigenvectors form an orthonormal basis. Therefore, the matrix $A$ is an orthonormal matrix.

Step 6: choosing principal components and forming a feature vector matrix. Eigenvectors associated with the biggest eigenvalues of the covariance matrix $C$ follow the most significant patterns of the data. Those eigenvectors are called the principle components of the data set. Therefore, the eigenvalues of the covariance matrix $C$ work as measures of how much information is contained in the principal components.

The feature vector matrix $A_q$ represented in Equation 2.6 is an $d \times q (q < d)$ matrix that contains only $q$ eigenvectors (principal components) from the matrix of eigenvectors $A$.

$$ A_q = [v_1, v_2, v_3, \ldots, v_q] \quad (2.6) $$

In order to select the principal components that form the feature vector matrix $A_q$, the variance contribution rate ($VCR$) and the total variance contribution rate ($TVC$) (they are proposed in the IEEE paper presented in Reference [2]) are computed as in Equation 2.7 and Equation 2.8 respectively. When the $TVC$ is significantly high then $q$ eigenvectors associated with the biggest $q$ eigenvalues can be selected.

$$ VCR_k (\%) = \frac{\lambda_k}{\sum_{k=1}^{d} \lambda_k} \times 100, \quad k = 1, \ldots, d \quad (2.7) $$

$$ TVC (\%) = \frac{\sum_{k=1}^{q} \lambda_k}{\sum_{k=1}^{d} \lambda_k} \times 100, \quad q = 1, \ldots, d \quad (2.8) $$

Step 7: performing the principal components transform (also called the Hotelling or Karhunen-Loéve transform).

Equation 2.9 is used for projecting the data on the eigenspace. The columns of the matrix $Y$ represent the coordinates of the projected data in the eigenspace.

$$ Y = A_q^T R \quad (2.9) $$

The mean of the matrix $Y$ is $m_Y = E [A_q^T R] = A_q^T E [R] \cong 0$. This has important meaning in face recognition. In fact, $Y$ gives the original centered data solely in terms of the selected principal components instead of the original axes. It is possible to express data in terms of any two perpendicular axes as shown in Reference [6], Page 167, Theorem 5.7.

Finally, by projecting the centered data on the eigenspace, the PCA algorithm is completely done.
2.2.2.3 An Example of the PCA Algorithm

The example moves simultaneously with the PCA steps illustrated in Sub-subsection 2.2.2.2 until a data set is transformed as follows:

Step 1: the two-dimensional data set $D$ shown in Equation 2.10 is obtained for performing the PCA algorithm. The plot of the data is shown in Figure 2.1.

\[
D = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 1 & 5 & 2 & 6 & 5 & 10 & 7 & 11 & 8 \end{bmatrix}. \quad (2.10)
\]

Figure 2.1: The plot of the data set $D$.

Step 2: the mean vector $m_D$ of the data set is computed as in Equation 2.11.

\[
m_D = \frac{1}{10} \sum_{k=1}^{10} D_k = \begin{bmatrix} 5.50 \\ 5.70 \end{bmatrix}. \quad (2.11)
\]

Step 3: the mean is subtracted from each of the data dimensions then the centered data set $R$ is obtained as in Equation 2.12. The plot of the centered data is shown in Figure 2.2.

\[
R = [D_1 - m_D, \ldots, D_{10} - m_D]
= \begin{bmatrix} -4.50 & -3.50 & -2.50 & -1.50 & -0.50 & 0.50 & 1.50 & 2.50 & 3.50 & 4.50 \\ -3.70 & -4.70 & -0.70 & -3.70 & 0.30 & -0.70 & 4.30 & 1.30 & 5.30 & 2.30 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}. \quad (2.12)
\]
Step 4: the covariance matrix $C$ is computed as in Equation 2.13. Since the non-diagonal elements of the covariance matrix are positive then both the $x$ and $y$ variables are expected to increase together.

\[
C = \frac{1}{10 - 1} \sum_{k=1}^{10} (D_k - m_D)(D_k - m_D)^T \\
= \frac{1}{10 - 1} \times R R^T \\
= \begin{bmatrix} 9.1667 & 8.7222 \\ 8.7222 & 11.5667 \end{bmatrix}.
\]  

(2.13)

Step 5: the eigenvectors and eigenvalues of the covariance matrix $C$ are computed.
respectively as in Equation 2.14 and Equation 2.15.

\[
A = [v_1, v_2]
\]

\[
= \begin{bmatrix}
0.6572 & -0.7538 \\
0.7538 & 0.6572
\end{bmatrix}
\]  \hspace{1cm} (2.14)

\[
\Lambda = \begin{bmatrix}
\lambda_1 & 0 \\
0 & \lambda_2
\end{bmatrix}, \hspace{0.5cm} \text{where } \lambda_1 \geq \lambda_2
\]

\[
= \begin{bmatrix}
19.1711 & 0 \\
0 & 1.5623
\end{bmatrix}
\]  \hspace{1cm} (2.15)

The centered data as well as the orthonormal eigenvectors are plotted together in Figure 2.3. Figure 2.3 shows how the data have totally a noticed pattern; and as anticipated from the covariance matrix, the two variables are increasing together. The eigenvectors are plotted as diagonal dotted lines. As expected, they are perpendicular to each other; more importantly they highlight patterns in the data where the highly correlated eigenvector passes through the middle of the points. It divides the points to two sets, like drawing a line of the best fit; and it describes the most significant relationship between the data dimensions. The other eigenvector follows little patterns of the data.

Step 6: for choosing principal components and forming the feature vector matrix \(A_q\), the variance contribution rate \(VCR\) and the total variance contribution rate \(TVC\) are calculated in Table 2.1. Based on Table 2.1, the feature vector matrix in Equation 2.16 only contains the eigenvector which is associated with the biggest eigenvalue (the highly correlated eigenvector).

**Table 2.1: The calculations of the \(VCR\) and \(TVC\).**

<table>
<thead>
<tr>
<th>(k)</th>
<th>(\lambda_k)</th>
<th>(VCR_k) (%)</th>
<th>(TVC) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19.1711</td>
<td>92.4648</td>
<td>92.4649</td>
</tr>
<tr>
<td>2</td>
<td>1.5623</td>
<td>7.5352</td>
<td>100</td>
</tr>
</tbody>
</table>

\[
A_q = \begin{bmatrix}
0.6576 \\
0.7538
\end{bmatrix}
\]  \hspace{1cm} (2.16)

Therefore, by computing the eigenvectors of the covariance matrix \(C\) and selecting the highest correlated ones then lines that describe the data are extracted. The rest
Chapter 2. Theoretical Background

Figure 2.3: The plot of the centered data set $R$ as well as the orthonormal eigenvectors.

of the steps involve transforming the data such that they are expressed in terms of the extracted lines.

Step 7: Equation 2.9 is used for performing the principal components transform; and the coordinates of the projected data in the eigenspace are shown in Equation 2.17.

$$Y^T = \begin{bmatrix}
\text{Along } \mathbf{v}_1 \text{ axis} \\
-5.7461 \\
-5.8427 \\
-2.1705 \\
-3.7746 \\
-0.1025 \\
-0.1991 \\
4.2269 \\
2.6228 \\
6.2950 \\
4.6908
\end{bmatrix}.$$ (2.17)

In Equation 2.17, it can be seen that the dimensions of the projected data are reduced because the highest correlated eigenvector is only selected and the lowest one is neglected then some information is lost here. The projected centered data are plotted as in Figure 2.4. As shown in Figure 2.4, the projected centered data represent
Section 2.2. Digital Processing

a series of data items along the highest correlated eigenvector axis $v_1$ without any information about the data along the axis of the lowest correlated eigenvector $v_2$.

![Figure 2.4: The projected centered data by using just the highest correlated eigenvector $v_1$.](image)

Therefore, in this step, the data are expressed in terms of the patterns between them where the patterns are the extracted lines that highly characterize the relationships between the data.

2.2.2.3.1 Taking All Eigenvectors as Principal Components for Doing PCA

We want to figure out what happens in the example presented in Sub-subsection 2.2.2.3 when all eigenvectors are selected as principal components?

The coordinates of the projected data in the eigenspace when all eigenvectors are taken to form the feature vector matrix $A_q$ (i.e. $A = A_q$) are shown in Equation 2.18; and they are plotted in Figure 2.5. The projected data in Equation 2.17 is exactly equal to the first dimension of the projected data in Equation 2.18. The plot in Figure 2.5 and the plot of the original centered data in Figure 2.2 are typically the
same except that in Figure 2.5 the eigenvectors are the axes instead of $x$ and $y$ axes.

\[
Y^T = \begin{bmatrix}
\text{Along } v_1 \text{ axis} & \text{Along } v_2 \text{ axis} \\
-5.7461 & 0.9604 \\
-5.8427 & -0.4505 \\
-2.1705 & 1.4244 \\
-3.7746 & -1.3008 \\
-0.1025 & 0.5740 \\
-0.1991 & -0.8369 \\
4.2269 & 1.6951 \\
2.6228 & -1.0301 \\
6.2950 & 0.8448 \\
4.6908 & -1.8805 \\
\end{bmatrix}.
\] (2.18)

**Figure 2.5:** The projected centered data when all eigenvectors are used as principal components.

Therefore, there is no loss of information when all eigenvectors are selected as principal components for doing PCA.

**2.2.2.4 Reconstruction of the Data Set $D$**

For reconstructing the data set, Equation 2.9 is turned around to get the centered data set $R$ as in Equation 2.19. Then the mean vector $m_D$ is added again to obtain
the reconstructed data set $\hat{D}$ as in Equation 2.20.

\[
R = \left(A_{q}^{T}\right)^{-1} \times Y \quad (2.19)
\]

\[
\hat{D} = \left(A_{q}^{T}\right)^{-1} \times Y + [m_{D}, \ldots, m_{D}]_{d \times n} \quad (2.20)
\]

Since $A_{q}^{T}$ in Equation 2.20 is not a square matrix and it has orthonormal (implies orthogonality) column vectors then the left inverse (Reference [6], Page 21, Definition 1.11) can be used to obtain its inverse. Its inverse is found to be $\left(A_{q}^{T}\right)^{-1} = \left(A_{q}^{T}\right)^{T} = A_{q}$. Then Equation 2.20 can be simplified as in Equation 2.21.

\[
\hat{D} = A_{q} \times Y + [m_{D}, \ldots, m_{D}]_{d \times n} \quad (2.21)
\]

2.2.2.5 Reconstruction of the Data Set $D$ of the Example in Sub-subsection 2.2.2.3

Equation 2.21 is used to reconstruct the data set $D$ of the example in Sub-subsection 2.2.2.3 as in Equation 2.22. The reconstructed data are plotted in Figure 2.6. As seen in Figure 2.6, some information is lost from the reconstructed data due to some of the eigenvectors are used as principal components in performing PCA transform.

\[
\hat{D} = \begin{bmatrix}
    \text{Along } x \text{ axis} & \text{Along } y \text{ axis} \\
    1.7239 & 1.3689 \\
    1.6605 & 1.2960 \\
    4.0736 & 4.0640 \\
    3.0195 & 2.8549 \\
    5.4327 & 5.6228 \\
    5.3692 & 5.5500 \\
    8.2777 & 8.8860 \\
    7.2236 & 7.6769 \\
    9.6368 & 10.4449 \\
    8.5826 & 9.2357 \\
\end{bmatrix} \quad (2.22)
\]

2.2.2.5.1 Using All Eigenvectors for Reconstructing the Data Set $D$

When all eigenvectors are selected as principal components for performing PCA in the example in Sub-subsection 2.2.2.3 then the original data set $D$ will be reconstructed perfectly (i.e. $\hat{D} = D$) without loss of information.
2.2.2.6 Application of the PCA Algorithm to Images

The reason beyond performing the PCA algorithm on a simple database is to be able to provide plots of data for showing PCA behavior at each step. After demonstrating the PCA algorithm on a simple database, the idea can be generalized to see how the PCA algorithm works when the data set is composed of images. This idea is based on References [7] and [8]; and Reference [9], Section 12.5. The PCA algorithm can be applied to images as in the following steps:

Step 1: creating a database.
The database is composed of $n$, $N \times N$ images of faces (training faces) on black backgrounds such that $I_k$ where $k = 1, \ldots, n$; and $n$ is the total number of the training faces.

For decreasing the error rates of face detection and recognition, all face projections must be defined in the database; images must have the same size; images must have only faces and they are expanded to the boarders of the images; finally, images must have unified backgrounds in order to discriminate between the pixels occupying the backgrounds and the pixels occupying the faces.

Step 2: normalizing each training face $I_k$.
This normalization is for removing lighting effects on the training faces. It is very important to increase the accuracy of face recognition but it does not affect face
Section 2.2. Digital Processing

detection.

The normalization must be done just for the pixels occupying the faces to keep variations among the images just in the faces without the effects of the backgrounds. In order to block the pixels occupying the backgrounds for all training faces, a threshold must be picked to distinguish between the pixels occupying the faces and the pixels occupying the backgrounds. Note that, a number zero can not be taken to be the threshold although the training faces have black backgrounds because the MATLAB software does not read a black color exactly zero then some error will occur.

According to the definition of an image histogram (Reference [10], Section 3.3), if a training face has a unified background then the biggest histogram of the intensity levels will be for the pixels occupying the backgrounds because pixels that have the same intensity levels are the pixels occupying the backgrounds of the training face. Therefore, the threshold can be selected based on the average histogram for all training faces.

Step 3: centering each training face $I_k$.

Since operations are performed on two-dimensional images then images must be centered before centering the whole database. This can be done by simply subtracting the mean of the pixels occupying the face for a training face from each pixel on the face as in Equation 2.23 where $I_{C_k}$ is the $k^{th}$ centered training face; and $m_{f_k}$ is the mean of the pixels occupying the face for the $k^{th}$ training face. By doing that the pixels on the face will have zero mean that means the face is centered.

$$I_{C_k} = I_k - m_{f_k}, \quad \text{where } k = 1, \ldots, n$$

(2.23)

Step 4: representing each centered training face $I_{C_k}$ as a column image vector $\Gamma_k$.

Each $N \times N$ centered training face is represented as an $N^2$ column image vector by transposing the rows of pixels then stacking them one after another to form a column vector as in Equation 2.24.

$$\Gamma_k = \begin{bmatrix} Row_1^T \\ Row_2^T \\ \vdots \\ Row_N^T \end{bmatrix}$$

(2.24)

Step 5: calculating the average training face vector $\Psi$ as in Equation 2.25.

$$\Psi = \frac{1}{n} \sum_{k=1}^{n} \Gamma_k$$

(2.25)

Step 6: centering the set of the training faces.

The set of the training faces is centered by simply subtracting the mean training face vector $\Psi$ from each centered training face vector $\Gamma_k$ as in Equation 2.26 where $\Phi_k$ is the $k^{th}$ centered (with respect to the set of the training faces) training face. By doing
that the set of the training faces will have zero mean which means the database is centered.

\[ \Phi_k = \Gamma_k - \Psi, \quad \text{where } k = 1, \ldots, n \]  

(2.26)

Step 7: calculating a covariance matrix for all training faces as in Equation 2.27 where \( R = [\Phi_1, \Phi_2, \ldots, \Phi_n] \). The covariance matrix can be equal to \( RR^T \) due to \( \frac{1}{n} \) can be removed or left because it is just a normalization factor which affects all values by the same amount.

\[ C = \frac{1}{n} \sum_{k=1}^{n} \Phi_k \Phi_k^T = RR^T \]  

(2.27)

Step 8: computing the eigenvectors and eigenvalues of the covariance matrix \( C \).

The covariance matrix \( C \) is \( N^2 \times N^2 \) matrix where \( N^2 \) is the total number of pixels along one of its dimensions. The covariance matrix is usually too big which makes the calculation of the eigenvalues and eigenvectors is very difficult if not impossible. Hence, it is not practical to calculate the eigenvalues and eigenvectors for the such matrix but we will calculate them in this work for examining the performance of the PCA and IPCA algorithms.

The dimensions of the covariance matrix can be reduced to the number of the training faces. Let’s suppose the \( n \times n \) matrix \( R^T R \); the eigenvectors and eigenvalues of this matrix are found as in Equation 2.28 where \( u_k \) is the \( k^{th} \) eigenvector of the matrix \( R^T R \); and \( \mu_k \) is the \( k^{th} \) eigenvalue.

\[ R^T R u_k = \mu_k u_k, \quad \text{where } k = 1, \ldots, n \]  

(2.28)

The relationship between the eigenvector \( v_k \) of the matrix \( RR^T \) and the eigenvector \( u_k \) of the matrix \( R^T R \) can be obtained as in Equation 2.29.

\[ R^T R u_k = \mu_k u_k \]

\[ RR^T R u_k = \mu_k R u_k \]

\[ CRu_k = \mu_k R u_k \]

\[ Cv_k = \mu_k v_k, \quad \text{where } v_k = Ru_k \]  

(2.29)

Equation 2.29 implies a couple of important notes are \( RR^T \) can have up to \( N^2 \) eigenvalues and eigenvectors; \( R^T R \) can have up to \( n \) eigenvalues and eigenvectors; and \( n \) eigenvectors of the matrix \( RR^T \) associated with the biggest eigenvalues are exactly identical to the eigenvectors of the matrix \( R^T R \) and they are related as, \( v_k = Ru_k \).

The generated eigenvectors by using the reduced covariance matrix must be normalized. The normalization can be performed by dividing the vector \( v_k \) by its length such that \( \frac{v_k}{\|v_k\|} \) then the length of the normalized eigenvector \( v_k \) will be equal to one;
i.e., \( \|v_k\| = 1 \). From now on, the reduced covariance matrix is going to be used for calculating desired eigenvectors and eigenvalues.

Step 9: selecting principal components and forming a feature vector matrix.

Here, principal components are called eigenfaces. They constitute the calculated eigenvectors associated with the biggest eigenvalues.

The feature vector matrix \( A_q \) represented in Equation 2.30 is an \( N^2 \times q \) matrix that contains only \( q \) eigenvectors (principal components) such that \( q \ll N^2 \). Since the number of calculated eigenvectors by using the reduced covariance matrix \( R^T R \) is equal to the total number of the training faces then \( q = n \).

\[
A_q = [v_1, v_2, \ldots, v_q], \quad \text{where } q \leq n \tag{2.30}
\]

In order to select the eigenfaces that form the feature vector matrix \( A_q \), the variance contribution rate (\( VCR \)) and the total variance contribution rate (\( TVC \)) (they are proposed in the IEEE paper presented in Reference [2]) are computed as in Equation 2.31 and Equation 2.32 respectively. When the \( TVC \) is significantly high then \( q \) eigenvectors associated with the biggest \( q \) eigenvalues can be selected.

\[
VCR_k(\%) = \frac{\lambda_k}{\sum_{k=1}^{n} \lambda_k} \times 100, \quad k = 1, \ldots, n \tag{2.31}
\]

\[
TVC(\%) = \frac{\sum_{k=1}^{q} \lambda_k}{\sum_{k=1}^{n} \lambda_k} \times 100, \quad q = 1, \ldots, n \tag{2.32}
\]

Step 10: performing the principal components transform (also called the Hotelling or Karhunen-Loève transform).

Equation 2.33 is used for projecting the training faces on the eigenspace. The columns of the matrix \( Y \) represent the coordinates of the projected training faces in the eigenspace; and \( \Omega^n \) contains the coordinates of the \( n \)th projected training face.

\[
Y = A_q^T R = [\Omega^1 \ldots \Omega^n], \quad \text{where } \Omega^n = \begin{bmatrix} u_1^n \\ \vdots \\ u_q^n \end{bmatrix} \tag{2.33}
\]

Finally, by projecting the training faces on the eigenspace, the PCA algorithm is completely done.

### 2.2.2.7 Reconstruction of the Original Images

For reconstructing the training faces, Equation 2.33 is turned around to get the centered training faces matrix \( R \) as in Equation 2.34. The average training face vector \( \Psi \) is added again to obtain the reconstructed centered training faces vectors as in Equation 2.35 where \( \hat{\Gamma}_n \) is the \( n \)th reconstructed centered training face vector.
Also, the means of the pixels occupying the faces are added to get the reconstructed training faces vectors as in Equation 2.36 where \( \hat{I}_n \) is the \( n \)th reconstructed training face vector. Finally, \( \hat{I}_n \) can be represented in the same manner as in Step 4, Sub-subsection 2.2.2.6; to obtain the reconstructed training face \( \hat{I}_n \).

\[
R = A_q Y
\]

\[
[\hat{\Gamma}_1 \ldots \hat{\Gamma}_n] = A_q Y + [\Psi \ldots \Psi]_{N^2 \times n} \tag{2.35}
\]

\[
[\hat{I}_1 \ldots \hat{I}_n] = A_q Y + [\Psi \ldots \Psi]_{N^2 \times n} + [m_{f_1} \ldots m_{f_n}] \tag{2.36}
\]

2.2.3 An Improved Principal Components Analysis (IPCA) Algorithm

2.2.3.1 Introduction

IPCA is another statistical tool for identifying patterns in data. It is typically like PCA except in the way of selecting eigenvectors that form the feature vector matrix \( A_q \). It affords a new accurate method to measure the information content of principal components based on the information theory for improving PCA. IPCA acts better than PCA in the most of applications.

2.2.3.2 Analysis of the IPCA Algorithm

In order to estimate the degree of information content of eigenvectors, the concepts of Shannon information theory are fully used then two new concepts called the possibility information function (PIF) and the possibility information entropy (PIE) are obtained.

Eigenvalues can be transformed as in Equation 2.37 where \( d \) is the number of the dimensions of the data set. In Equation 2.37, it can be seen that \( 0 \leq \rho_k \leq 1 \), where \( k = 1, \ldots, d \). Therefore, \( \rho_k \) has the numerical properties of probability. Being similar with the definition of entropy, the PIF and PIE can be defined respectively as in Equation 2.38 and Equation 2.39. In Equation 2.39, \( H(T) \) reflects the unevenness of \( \rho_k \). According to the PIF and PIE, it can be obtained that firstly \( I(\lambda_k) \) denotes the information content included by \( \lambda_k \) where the bigger \( \lambda_k \) is associated with the bigger \( I(\lambda_k) \); Secondly, when all \( \rho_k \) are equal (i.e. uniformly distributed) then the PIE reaches its maximum.

\[
\rho_k = 1 - \frac{\lambda_k}{\sum_{k=1}^{d} \lambda_k}, \quad k = 1, \ldots, d \tag{2.37}
\]

\[
I(\lambda_k) = -\log_2 \rho_k, \quad k = 1, \ldots, d \tag{2.38}
\]

\[
H(T) = H(\rho_1, \ldots, \rho_d) = -\sum_{k=1}^{d} \rho_k \log_2 \rho_k \tag{2.39}
\]
For measuring the degree of information content of eigenvectors, two new concepts are used. The first one is the information rate ($IR$) shown in Equation 2.40; and the second one is the accumulated information rate ($AIR$) shown in Equation 2.41. When the $AIR$ is significantly high then $q$ eigenvectors associated with the biggest $q$ eigenvalues can be selected.

\[
IR_k(\%) = \frac{I(\lambda_k)}{\sum_{k=1}^{d} I(\lambda_k)} \times 100, \quad k = 1, \ldots, d \tag{2.40}
\]

\[
AIR(\%) = \frac{\sum_{k=1}^{q} I(\lambda_k)}{\sum_{k=1}^{d} I(\lambda_k)} \times 100, \quad q = 1, \ldots, d \tag{2.41}
\]

### 2.2.3.3 Applying IPCA on the Example in Sub-subsection 2.2.2.3

The $IR$ and $AIR$ are computed for the eigenvectors of the covariance matrix $C$ in the example in Sub-subsection 2.2.2.3; and the results are shown in Table 2.2. Based on Table 2.2, the feature vector matrix $A_q$ in Equation 2.42 only contains the eigenvector which is associated with the biggest eigenvalue (the highly correlated eigenvector).

**Table 2.2**: The calculations of the $IR$ and $AIR$.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$\lambda_k$</th>
<th>$\rho_k$</th>
<th>$I(\lambda_k)$</th>
<th>$IR_k(%)$</th>
<th>$AIR(%)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19.1711</td>
<td>0.0754</td>
<td>3.7293</td>
<td>97.0616</td>
<td>97.0616</td>
</tr>
<tr>
<td>2</td>
<td>1.5623</td>
<td>0.9247</td>
<td>0.1129</td>
<td>2.9384</td>
<td>100</td>
</tr>
</tbody>
</table>

\[
A_q = \begin{bmatrix}
0.6576 \\
0.7538
\end{bmatrix}.
\tag{2.42}
\]

### 2.2.4 Comparison of the PCA and IPCA Algorithms

Comparison between the PCA and IPCA algorithms is based on the values of the $TVC$ and $AIR$ that determine the selected eigenvectors for the feature vector matrix $A_q$ in Sub-subsection 2.2.2.3, Step 6; and in Sub-subsection 2.2.3.3.

The computed $TVC$ and $AIR$ are respectively equal to 92.4649% and 97.0616%. The $AIR$ is big enough but the $TVC$ is slightly small. Then the $AIR$ is providing us with more confidence to pick the eigenvector which is associated with the biggest eigenvalue but the $TVC$ is not. Therefore, the $AIR$ tells us more about information contained in the eigenvectors of the covariance matrix $C$. 
2.2.5 Applications of the PCA and IPCA Algorithms

2.2.5.1 Introduction

PCA and IPCA are applied in many fields. They have acceptable performance in many of them. In this subsection, the applications of the PCA and IPCA algorithms in image compression, face recognition and image detection are covered.

2.2.5.2 Image Compression

When some of the eigenvectors that are calculated from the covariance matrix \( C \) for all training faces are selected to form the feature vector matrix \( A_q \) then the dimensions of the reconstructed data set will be reduced. This implies that the PCA and IPCA algorithms work as compression. The algorithms are said to be lossy because a decompressed image is not exactly the same as the original one, but is generally worse.

2.2.5.3 Face Recognition

The PCA and IPCA algorithms are used to recognize an unknown face image based on the database which contains the training faces. For doing face recognition, an unknown face image is taken. It must have the same properties of the training faces. Hence, it must have the same size as the training faces; it has the same background; it has a projection as one of the training faces; finally, it has only face and it is expanded to the boarders of the image.

A couple of the operations explained in Sub-subsection 2.2.2.6 are applied to the unknown face image. The pixels occupying the face of the unknown face image are normalized and centered as in Step 2 and Step 3 respectively. The centered unknown face image is represented as a column image vector as in Step 4. The centered unknown image vector is centered in the set of the training faces by simply subtracting the mean training face vector \( \Psi \) from it as in Step 6. Then the unknown face image is projected on the eigenspace by using the PCA or IPCA feature vector matrix as in Step 10. The coordinates of the projected unknown face image in the eigenspace are shown in Equation 2.43.

\[
\Omega^{\text{Unknown}} = \begin{bmatrix} w_1^{\text{Unknown}} \\ \vdots \\ w_q^{\text{Unknown}} \end{bmatrix}
\]  
(2.43)

The Euclidean distance between the coordinates of the projected unknown face image and the coordinates of each training face is computed as in Equation 2.44. \( d_k \) is the distance between the coordinates of the projected unknown face image \( \Omega^{\text{Unknown}} \) and the coordinates of the \( k^{th} \) projected training face \( \Omega^k \); and \( n \) is the total number of the training faces. Note that, the distances between the unknown face image and
the training faces are measured along the new axes derived from the PCA algorithm but not along the original axes. It turns out that these axes work much better for recognizing faces because PCA has given us the original training faces in terms of the differences and similarities between them.

\[ d_k = \|\Omega_{\text{Unknown}} - \Omega^k\|, \quad \text{where } k = 1, \ldots, n \tag{2.44} \]

Then recognition can be performed by using Condition 2.1 where \( md_k \) is the minimum distance between the coordinates of the projected unknown face image \( \Omega_{\text{Unknown}} \) and the coordinates of the \( k^{th} \) projected training face \( \Omega^k \).

**Condition 2.1.** If,

\[ md_k = \min ([d_1, \ldots, d_n]), \quad \text{where } k \text{ can be any number between } 1 \text{ to } n \]

Then the unknown face image is recognized as the \( k^{th} \) training face; otherwise, it is an unknown face image.

At this point, it can be answered why the normalization of the training faces as well as the centering of the database and faces increase the accuracy of face recognition. That because when the database and faces are centered, the set of the projected vectors in the eigenspace will have zero mean; that means the vectors begin from the same origin; then the distance between a training face and its corresponding unknown face image will be very small compared with other training faces. Regarding the normalization of the training faces, when a training face and its corresponding unknown face image are normalized (i.e. they have the same length), the distance between them will decrease as shown in Figure 2.7. On the other hand, if they are unnormalized (i.e. they do not have the same length), the distance between them will increase as shown in Figure 2.8.

![Figure 2.7: A normalized training face and its normalized corresponding unknown face image.](image)
2.2.5.3.1 Setting up a Recognition Threshold

Unfortunately, when the unknown face image does not have a similar training face then getting the minimum distance \( md_k \) does not always mean that the unknown face image is recognized as the \( k^{th} \) training face. Therefore, a certain threshold must be used to increase the accuracy of recognition.

For setting up a recognition threshold, some different images are taken for each training face. These images are called tested images. They are known here just for picking the threshold. The distances between each training face and its corresponding tested images are obtained; and they are stacked in the row vectors \( T_1, \ldots, T_k, \ldots, T_n \), where \( T_k \) contains the smallest expected distances because the \( k^{th} \) training face and its corresponding tested images have the same person and the same face projection.

Thereafter, the mean \( m_k \) and standard deviation \( STD_k \) (the average distance from the mean to a point) are computed for each row vector \( T_k \) of the smallest distances. By calculating the means and the standard deviations, a certain threshold is established for each training face.

Then the recognition threshold can be applied to recognize the unknown face image by using Condition 2.2.

**Condition 2.2.** If,

\[
    m_k - STD_k \leq md_k \leq m_k + STD_k, \quad \text{where } k = 1, \ldots, n
\]

Then the unknown face image is recognized as the \( k^{th} \) training face; otherwise, it is an unknown face image.
2.2.5.4 Image Detection

The PCA and IPCA algorithms are used to detect if an unknown image contains a face or not based on a determined threshold for detection. Hence, in image detection, only a detection threshold is needed.

To obtain the detection threshold, tested images are generated from the database of the training faces in the same manner as in Sub-sub-subsection 2.2.5.3.1. From Sub-subsection 2.2.2.6, the preprocessing operations illustrated in Step 2, Step 3, Step 4, Step 6 and Step 10 are respectively applied to the tested images. Then the projected centered (with respect to the set of the training faces) tested images are reconstructed again without adding the average training face neither the means of the pixels occupying the faces. Each reconstructed tested image is normalized and multiplied by the biggest intensity from its centered tested image; this is done in order to make each centered tested image and its reconstruction have approximately the same dynamic range.

The Euclidean distance between each centered tested image and its reconstruction is computed as in Equation 2.45 where \( a_k \) is the distance between the \( k \)th centered tested image \( \Phi_{Tested \, Im}^k \) and its reconstruction \( \hat{\Phi}_{Tested \, Im}^k \); as well as \( t \) is the total number of the tested images. Thereafter, the computed distances are placed in the row vector \( S \). The mean \( m_S \) and the standard deviation \( STD_S \) (the average distance from the mean to a point) are calculated for the vector \( S \). By calculating the mean and the standard deviation, the detection threshold is established.

\[
a_k = \| \Phi_{Tested \, Im}^k - \hat{\Phi}_{Tested \, Im}^k \|, \quad \text{where} \quad k = 1, \ldots, t \quad (2.45)
\]

For applying the detection threshold, the unknown image \( I_{Unknown}^{Unknown} \) is picked for detection. The Euclidean distance \( a_{Unknown}^{Unknown} \) between the centered unknown image \( \Phi_{Unknown}^{Unknown} \) and its reconstruction \( \hat{\Phi}_{Unknown}^{Unknown} \) is computed. Then the unknown image \( I_{Unknown}^{Unknown} \) can be detected by means of Condition 2.3.

Condition 2.3. If,

\[
 m_S - STD_S \leq a_{Unknown}^{Unknown} \leq m_S + STD_S
\]

Then the unknown image \( I_{Unknown}^{Unknown} \) is detected as a face image; otherwise, it is not a face image.

2.2.6 Performance Analysis of the PCA and IPCA Algorithms

2.2.6.1 Introduction

In studying performance, attention is paid to study how each application behaves when the eigenfaces that are generated by using the PCA algorithm are used; the
2.2.6.2 Analysis of Compression Performance

2.2.6.2.1 Introduction

Compression performance can be analyzed from three points of view are the speed of compression and reconstruction, the quality of a reconstructed image, and the size of compression. Each one is explained in details in this sub-subsection.

2.2.6.2.2 Speed of Compression and Reconstruction

The number of the eigenfaces that is used to compress and reconstruct the training faces mainly controls the processing speed of compression and reconstruction. When a small number of the eigenfaces is used to project and reconstruct the training faces then the processing speed will increase and vice versa.

2.2.6.2.3 Quality of a Reconstructed Image

For measuring the quality of a reconstructed image, the mean squared error (MSE) is computed as in Equation 2.46 to measure an error between the image $I_k$ and its reconstruction $\hat{I}_k$; where $N$ is the number of rows and columns of the image $I_k$.

$$e_{MSE} = \frac{1}{N^2} \sum_{r=1}^{N} \sum_{c=1}^{N} [\hat{I}_k(r, c) - I_k(r, c)]^2$$

(2.46)

2.2.6.2.4 Size of Compression

2.2.6.2.4.1 Introduction

The size of compression can be measured in two ways are an information rate and the mean squared error (MSE) of compressed images.

2.2.6.2.4.2 Information Rate

An information rate measures how much information is after compression compared with information before compression; in other words, it measures the number of pixels after compression compared with before. This can be accomplished as
Equation 2.47.

Before Compression : After Compression

\[
\begin{pmatrix}
\frac{n}{n \times N^2} \times \frac{N^2}{n \times N^2} \\
\text{The total number of the training faces} & \text{An image size}
\end{pmatrix} \times 100 : \begin{pmatrix}
\frac{N^2 \times q}{n \times N^2} + \frac{q \times n}{n \times N^2} + \frac{N^2}{n \times N^2} \\
\text{The feature vector matrix} A_q & \text{The coordinates matrix} Y \text{ of the projected training faces} & \text{The average training face vector} \Psi
\end{pmatrix} \times 100
\] (2.47)

In Equation 2.47 the rows and columns for the pixels occupying the faces as well as the means of the pixels on the faces are not considered in the overall size after compression because the operation of face centering is not really important in image compression; and it does not have any effect if it is done or not; but it has been done here because it is important for other applications. The normalization is done in Equation 2.47 to make the overall information before compression is equal to 100% all the time in order to make comparison easier.

### 2.2.6.2.4.3 Mean Squared Error (MSE) of Compressed Images

The mean squared error (MSE) between the exact and approximate reconstruction of the training face vector \(I_k\) is calculated as follows,

Equation 2.48 shows the exact reconstruction of the training face vector \(I_k\).

\[
I_k = \sum_{i=1}^{n} w_i^k v_i + \Psi
\] (2.48)

And, Equation 2.49 shows the approximate reconstruction of the training face vector \(I_k\).

\[
\hat{I}_k = \sum_{i=1}^{q} w_i^k v_i + \Psi
\] (2.49)
The error between $I_k$ and $\hat{I}_k$ can be computed as in Equation 2.50.

$$e = I_k - \hat{I}_k$$

$$= \sum_{i=1}^{n} w_i^k v_i + \Psi - \sum_{i=1}^{q} w_i^k v_i - \Psi$$

$$= \sum_{i=1}^{n} w_i^k v_i - \sum_{i=1}^{q} w_i^k v_i$$

$$= \sum_{i=q+1}^{n} w_i^k v_i$$  \hspace{1cm} (2.50)

For computing the MSE of the linear estimate $\hat{I}_k$, Equation 2.51 can be used.

$$E\left[ e^T e \right] = E \left[ \left( \sum_{i=q+1}^{n} w_i^k v_i \right)^T \left( \sum_{m=q+1}^{n} w_m^k v_m \right) \right]$$

$$= E \left[ \sum_{i=q+1}^{n} w_i^k v_i^T \sum_{m=q+1}^{n} w_m^k v_m \right]$$

$$= E \left[ \sum_{i=q+1}^{n} \sum_{m=q+1}^{n} w_i^k w_m^k v_i^T v_m \right]$$

$$= \sum_{i=q+1}^{n} \sum_{m=q+1}^{n} E \left[ w_i^k w_m^k \right] v_i^T v_m$$  \hspace{1cm} (2.51)

To find $E \left[ w_i^k w_m^k \right]$, a covariance matrix for the coordinates matrix $Y$ of the projected
training faces must be computed as in Equation 2.52

\[ C_Y = YY^T \]

\[ = [A^T R] [A^T R]^T \]

\[ = [A^T R] [R^T A] \]

\[ = A^T R R^T A \]

\[ = A^T CA \] \hspace{1cm} (2.52)

From Reference [3], Page 169, Theorem A.1.30; since \( C \) is a real and symmetric matrix as well as \( A \) is an orthonormal matrix then \( C_Y \) can be written as in Equation 2.53 where \( I \) is the identity matrix.

\[ C_Y = A^T CA \]

\[ = \underbrace{A^T A}_{I} \Lambda \underbrace{A^T A}_{I} \]

\[ = IAI \]

\[ = \Lambda \]

\[ = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & \lambda_n \end{bmatrix} \] \hspace{1cm} (2.53)

Hence, from Equation 2.53, it can be concluded that,

\[ E[w_i^k w_m^k] = \begin{cases} \lambda_i & \text{when } i = m \\ 0 & \text{when } i \neq m \end{cases} \]

And,

\[ v_i^T v_m = \begin{cases} 1 & \text{when } i = m \\ 0 & \text{when } i \neq m \end{cases} \] Because \( v_i \) and \( v_m \) form an orthonormal basis.
Therefore, when \( i = m \), the MSE between the exact and approximate reconstruction of the training face vector \( I_k \) is obtained as in Equation 2.54.

\[
E[e^T e] = \sum_{i=q+1}^{n} \sum_{m=q+1}^{n} E \left[ w_i^k w_m^k \right] v_i^T v_m
\]

\[
= \sum_{i=q+1}^{n} E \left[ (w_i^k)^2 \right] \cdot 1
\]

\[
= \sum_{i=q+1}^{n} \lambda_i
\]

(2.54)

2.2.6.3 Analysis of Recognition Performance

2.2.6.3.1 Introduction

Recognition performance can be analyzed from two points of view are the speed of recognition and an error rate. Each one is explained in details in this sub-subsection.

2.2.6.3.2 Speed of Recognition

The number of the selected eigenfaces that is used to recognize the unknown face image mainly controls the recognition speed. When the number of the selected eigenfaces decreases, the processing speed increases and vice versa.

2.2.6.3.3 Error Rate

The error rate computes the percentage of error in recognition. It can be computed as in Equation 2.55 where \( L \) is the total number of the tested images; \( SR \) is the total number of successes in the recognition of the tested images; \( FR \) is the total number of failures in the recognition of the tested images; and \( ER \) is the error rate.

\[
ER(\%) = \frac{FR}{L} \times 100
\]

(2.55)

2.2.6.4 Analysis of Detection Performance

2.2.6.4.1 Introduction

Detection performance can be analyzed from two points of view are the speed of detection and an error rate. Each one is explained in details in this sub-subsection.
2.2.6.4.2 Speed of Detection

The number of the selected eigenfaces that is used to detect the unknown image mainly controls the detection speed. When the number of the selected eigenfaces decreases, the processing speed increases and vice versa.

2.2.6.4.3 Error Rate

The error rate computes the percentage of error in detection. It can be computed as in Equation 2.56 where $L$ is the total number of the tested images; $SD$ is the total number of successes in the detection of the tested images; $FD$ is the total number of failures in the detection of the tested images; and $ER$ is the error rate.

$$ER(\%) = \frac{FD}{L} \times 100$$  \hspace{1cm} (2.56)

2.3 Analog Optical Information Processing

2.3.1 Introduction

Analog optical information processing is an important area which recalls the linearity concepts of imaging systems in order to synthesize an optical model that can perform one or multiple functions. The focus of this section is about providing a theoretical background for designing an optical model for object detection and face recognition. Concentration will be limited to coherent optical models for some reasons will be mentioned in the next subsection. This section is based on Reference [3], Chapter 8.

2.3.2 Coherent and Incoherent Optical Image Processing Systems

This subsection shows the difference between the usage of spatially incoherent light and the usage of spatially coherent light in optical information processing. Spatially incoherent light has some big advantages but on the other hand it has severe disadvantages. The advantages of spatially incoherent light are:

1. It is free from coherent artifacts such as dust specks on optical components and the speckle phenomenon.

2. Data can be introduced to a system by using incoherent light sources such as light-emitting diode (LED) arrays or cathode-ray tube (CRT) displays; but in coherent systems, complicated and costly spatial light modulators (SLMs) are used to introduce data.
3. In general, incoherent systems are easier than coherent systems in physical implementation.

On the other hand, spatially incoherent light has disadvantages that make us prefer to use spatially coherent light in our model. The disadvantages of spatially incoherent light are:

1. An incoherent optical system does not have a frequency plane but a coherent optical system has a plane at a distance \( f \) from a lens is called a frequency plane or a focal plane. The absence of this plane makes the manipulation of an input spectrum is very difficult rather than just the direct manipulation of a spectrum on a back focal plane.

2. Incoherent optical systems are linear in intensity. The manipulation of intensity in optical processing systems is very complex if it is not impossible because intensity is a positive and real physical quantity. For instance, there is no a normal optical method to subtract two intensity patterns; but coherent optical systems are linear in complex amplitude; consequently, if one wants to subtract two complex amplitude patterns then the patterns can be added together with a \( \pi \) radian phase shift between them.

3. The spectrum of an incoherent image that is generated from an incoherent optical system always has the biggest spectral component at the origin. This makes a produced incoherent image has low contrast. Therefore, incoherent optical systems need a huge use of electronics in order to enhance an output incoherent image and makes it comparable to an output coherent image.

Due to these serious disadvantages, spatially coherent light is going to be used in our model.

2.3.3 Coherent Optical Information Processing Systems

We now present the coherent optical information processing model used for object detection and face recognition. From the last subsection, it is known that coherent optical systems are linear in complex amplitude; therefore, filtering processes can be performed by direct manipulation of complex amplitude appearing in the back focal plane of a Fourier transforming lens. There are a large number of system architectures that can do frequency domain filtering but a pretty conceptually straightforward system shown in Figure 2.9 is implemented. This model for coherent optical information processing is called 4f model because a distance that separates the input plane \( P_1 \) from the output plane \( P_3 \) is composed of four separate distances of length \( f \). The length of this model from the point source \( S \) until the output plane is \( 5f \).

The collimating lens \( L_1 \) is used to collimate light from the point source \( S \). The input transparency is placed in the input plane \( P_1 \) against the collimating lens \( L_1 \).
The input plane is placed against the collimating lens in order to decrease the total length of the coherent model. The amplitude transmittance of the input transparency is \( g(x_1, y_1) \). The input transparency is illuminated by a uniform normally incident plane (or spherical) wave of amplitude \( A \). Then the complex amplitude distribution of the field just behind the input transparency (i.e. the field transmitted by the input transparency) is \( U_1(x_1, y_1) = Ag(x_1, y_1) \). The Fourier transforming lens \( L_2 \) is Fourier transforming the illuminated input transparency in its back focal plane \( P_2 \). A transparency is inserted in the back focal plane to modulate the amplitude transmittance over that plane. Then the complex amplitude distribution \( U_2(x_2, y_2) \) of the Fourier transformed field can be found as in Equation 2.57.

\[
U_2(x_2, y_2) = \frac{1}{j\lambda f} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U_1(x_1, y_1) \exp\left(-j\frac{2\pi}{\lambda f}(x_1 x_2 + y_1 y_2)\right) dx_1 dy_1
\]

\[
= \frac{A}{j\lambda f} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x_1, y_1) \exp\left(-j\frac{2\pi}{\lambda f}(x_1 x_2 + y_1 y_2)\right) dx_1 dy_1
\]

\[
= \frac{A}{j\lambda f} G\left(\frac{x_2}{\lambda f}, \frac{y_2}{\lambda f}\right)
\]

\[
= k_1 G\left(\frac{x_2}{\lambda f}, \frac{y_2}{\lambda f}\right)
\]

(2.57)

Where \( k_1 \) is a complex constant; \( \lambda \) is the light wavelength in meter (m); and \( G\left(\frac{x_2}{\lambda f}, \frac{y_2}{\lambda f}\right) \) is the Fourier spectrum of the Fourier transformed field.
A desired filter can be synthesized and placed in the plane $P_2$ in order to manipulate $G \left( \frac{x_2}{\lambda f}, \frac{y_2}{\lambda f} \right)$. Let the transfer function of a synthesized filter be represented by $H$ then the complex amplitude distribution in the back focal plane of the filter should be as in Equation 2.58 where $k_2$ is a complex constant.

$$U_f (x_2, y_2) = k_2 H \left( \frac{x_2}{\lambda f}, \frac{y_2}{\lambda f} \right)$$  \hspace{1cm} (2.58)

The spectrum of the field just behind the transparency of the plane $P_2$ is $G \left( \frac{x_2}{\lambda f}, \frac{y_2}{\lambda f} \right) H \left( \frac{x_2}{\lambda f}, \frac{y_2}{\lambda f} \right)$. The Fourier transforming lens $L_3$ Fourier transforms the altered spectrum in its back focal plane $P_3$. There is no an optical component does inverse Fourier transform; therefore, another Fourier transforming lens is used to produce the final complex amplitude distribution on the output plane $P_3$. The usage of the two consecutive Fourier transforming lenses makes the coordinates of the output plane $P_3$ inverted (Reference [11], Page 25, “Application of the Fourier Transform”). The coordinates inversion problem can be overcome by inverting the coordinates of the output plane $P_3$.

This coherent model has a disadvantage that vignetting can happen through performing the first Fourier transform. In order to overcome this problem, the input plane $P_1$ can be placed against the Fourier transforming lens $L_2$.

There are at least two methods for synthesizing the frequency plane filter for coherent optical systems. One of these methods is by using the joint transform correlator (JTC). This method is discussed in the next subsection.

### 2.3.4 The Joint Transform Correlator (JTC)

This correlator is used to synthesize the frequency plane filter in order to manipulate the spectrum on the back focal plane of the Fourier transforming lens $L_2$. This method was invented by Weaver and Goodman [4] and called as the joint transform correlator. The filter architecture is shown in Figure 2.10.

The filter is divided to two stages: recording the filter and getting the filter output. In the recording process, the lens $L_1$ collimates light from the point source $S$. Two input transparencies are placed in the input plane $P_1$. The first transparency is for the desired impulse response $h$ and centered at the coordinate $\left( 0, \frac{Y}{2} \right)$. The other transparency is for the data $g$ to be filtered and centered at the coordinate $\left( 0, -\frac{Y}{2} \right)$. Hence, their centers are separated by the distance $Y$. The input transparencies are illuminated by a uniform normally incident plane (or spherical) wave of amplitude $A$. Then the complex amplitude distribution of the field just behind the input transparencies (i.e. the field transmitted by the input transparencies) is obtained as in Equation 2.59.

$$U_1 (x_1, y_1) = A \left[ h \left( x_1, y_1 - \frac{Y}{2} \right) + g \left( x_1, y_1 + \frac{Y}{2} \right) \right]$$  \hspace{1cm} (2.59)
The Fourier transforming lens $L_2$ is Fourier transforming the field transmitted by the input transparencies in its back focal plane $P_2$. The complex amplitude distribution $U_2(x_2, y_2)$ of the Fourier transformed field is obtained as in Equation 2.60. The derivation of $U_2(x_2, y_2)$ is shown in Appendix E.

$$U_2(x_2, y_2) = \frac{A}{j\lambda f} \exp^{-j\frac{\pi}{\lambda f} y_2} H \left( \frac{x_2}{\lambda f}, \frac{y_2}{\lambda f} \right) + \frac{A}{j\lambda f} \exp^{j\frac{\pi}{\lambda f} y_2} G \left( \frac{x_2}{\lambda f}, \frac{y_2}{\lambda f} \right) \quad (2.60)$$

From linear algebra, if $Z_1$ and $Z_2$ are complex numbers; as well as $Z_1^*$ and $Z_2^*$ are their conjugates respectively then $|Z_1 + Z_2|^2 = |Z_1|^2 + |Z_2|^2 + (Z_1 Z_2^* + Z_2 Z_1^*)$. Using this operation on complex numbers and another operation is that the conjugate of the exponential function $\exp^{jx}$ is equal to $\exp^{-jx}$ (conversely, the conjugate of the exponential function $\exp^{-jx}$ is equal to $\exp^{jx}$) then the incident intensity on the back
Chapter 2. Theoretical Background

focal plane of the lens \( L_2 \) is computed as in Equation 2.61.

\[
I(x_2, y_2) = \frac{A^2}{\lambda^2 f^2} \left[ \left| H \left( \frac{x_2}{\lambda f}, \frac{y_2}{\lambda f} \right) \right|^2 + \left| G \left( \frac{x_2}{\lambda f}, \frac{y_2}{\lambda f} \right) \right|^2 + \right.
\]

\[
+ H \left( \frac{x_2}{\lambda f}, \frac{y_2}{\lambda f} \right) G^* \left( \frac{x_2}{\lambda f}, \frac{y_2}{\lambda f} \right) \exp^{-j \frac{2\pi}{\lambda} Y y_2} + \right.
\]

\[
+ H^* \left( \frac{x_2}{\lambda f}, \frac{y_2}{\lambda f} \right) G \left( \frac{x_2}{\lambda f}, \frac{y_2}{\lambda f} \right) \exp^{j \frac{2\pi}{\lambda} Y y_2} \right] \quad (2.61)
\]

Note that, the recorded transparency in the plane \( P_2 \) is supposed to have an amplitude transmittance that is proportional to the intensity \( I(x_2, y_2) \). To obtain the output of the filter, the recorded transparency is illuminated by a uniform normally incident plane (or spherical) wave of amplitude \( B \). The complex amplitude distribution of the field just behind the recorded transparency (i.e. the field transmitted by the recorded transparency) is \( U_r(x_2, y_2) = BI(x_2, y_2) \). The lens \( L_4 \) is Fourier transforming the transmitted field in its back focal plane \( P_3 \). The complex amplitude distribution of the Fourier transformed field in the output plane \( P_3 \) is computed in Equation 2.62.

\[
U_3(x_3, y_3) = \mathcal{F} \{ U_r(x_2, y_2) \}
\]

\[
= B \mathcal{F} \{ I(x_2, y_2) \} \quad (2.62)
\]

From linear algebra, if \( Z_1 \) is a complex number and \( Z_1^* \) is its conjugate then \( |Z_1|^2 = Z_1 Z_1^* \). Using this operation on complex numbers; and using another properties and theorems of Fourier transform are the convolution theorem (Reference [12], Page 37, Property 12), the complex conjugation property (Reference [12], Page 28, Property 3), and the property that the Fourier transform of the shifted impulse response \( \delta(t - t_o) \) is equal to \( \exp^{-j 2\pi ft_o} \) where \( t_o \) is the amount of the shift; then the field in the back focal plane \( P_3 \) is found as in Equation 2.63.

\[
U_3(x_3, y_3) = B \frac{A^2}{\lambda^2 f^2} [h(x_3, y_3) \otimes h^*(-x_3, -y_3) + g(x_3, y_3) \otimes g^*(-x_3, -y_3) +
\]

\[
+ h(x_3, y_3) \otimes g^*(-x_3, -y_3) \otimes \delta(x_3, y_3 - Y) +
\]

\[
+ h^*(-x_3, -y_3) \otimes g(x_3, y_3) \otimes \delta(x_3, y_3 + Y)] \quad (2.63)
\]

The complex distribution is composed of four terms. The first two terms respectively represent the cross-correlation of the impulse response \( h \) and itself as well as the
cross-correlation of the data \( g \) and itself. The third and fourth terms represent the cross-correlations of \( h \) and \( g \). The third and fourth terms are typically the same except that the third cross-correlation is centered at \((0, Y)\) and the fourth term is centered at \((0, -Y)\). There is not too much thing to do with the first and second terms but the third and fourth terms are of interest. The cross-correlations of \( h \) and \( g \) can be written as in Equation 2.64 and Equation 2.65.

\[
h(x_3, y_3) \otimes g^* (-x_3, -y_3) \otimes \delta (x_3, y_3 - Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(\xi, \eta) g^*(\xi - x_3, \eta - (y_3 - Y)) d\xi d\eta
\]

\[
= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(\xi, \eta) g^*(\xi - x_3, \eta - y_3 + Y) d\xi d\eta \tag{2.64}
\]

Note that,

\[
h(x_3, y_3) \otimes g^* (-x_3, -y_3) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(\xi, \eta) g^*(\xi - x_3, \eta - y_3) d\xi d\eta
\]

Similarly,

\[
h^* (-x_3, -y_3) \otimes g(x_3, y_3) \otimes \delta (x_3, y_3 + Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(\xi, \eta) h^*(\xi - x_3, \eta - y_3 - Y) d\xi d\eta \tag{2.65}
\]

From digital signal processing (DSP), the main difference between cross-correlation and convolution is that in cross-correlation the functions \( g^* (-x_3, -y_3) \) and \( h^* (-x_3, -y_3) \) are not rotated by \( 180^\circ \) before doing the cross-correlation process but in convolution they must be rotated by \( 180^\circ \) before doing a convolution process. To get the convolution of the impulse response \( h \) and the data \( g \), the input transparency of \( h \) or \( g \) (just one of them) in the recording stage must be rotated by \( 180^\circ \) along the spatial coordinate \( x_1 \) as well as along the spatial coordinate \( y_1 \). If the input transparency of the impulse response \( h \) is rotated then it will be \( h(-x_1, -y_1 + \frac{Y}{2}) \) instead of \( h(x_1, y_1 - \frac{Y}{2}) \); similarly, if the input transparency of the data \( g \) is rotated then it will be \( g(-x_1, -y_1 - \frac{Y}{2}) \) instead of \( g(x_1, y_1 + \frac{Y}{2}) \).

From digital signal processing (DSP), the bandwidth of a resulting function from the convolution, correlation or cross-correlation of two functions is equal to the sum of their bandwidths [13]. For example, if the bandwidths of the functions \( m(t) \) and \( f(t) \) are three and one respectively; then the bandwidth of their convolution is equal to four. Therefore, the bandwidths of the patterns of the cross-correlated field in the output plane \( P_3 \) along the spatial coordinates \( x_3 \) and \( y_3 \) are illustrated in Figure 2.11.

Obviously from Figure 2.11, in order to prevent overlapping between the patterns of the cross-correlated field that are centered at \((0, 0)\), \((0, Y)\) and \((0, -Y)\) (i.e. they are fully separated); the distance between the centers of the input transparencies must satisfy Relation 2.1 where \( W_g \) and \( W_h \) are the widths of \( g \) and \( h \) respectively in the
Figure 2.11: The bandwidths of the patterns of the cross-correlated field in the output plane $P_3$ along the spatial coordinates $x_3$ and $y_3$.

direction of the $y$-coordinate. It is really important to be noticed that in Figure 2.11, the highest cross-correlation exists in the center of each cross-correlated pattern and decreases by moving away from the center.

Relation 2.1.

$$Y > \max \{W_h, W_g\} + \frac{W_g}{2} + \frac{W_h}{2}$$

The constant $c$ is added to the distance $Y$ for realizing Relation 2.1. In order to confine the patterns of the cross-correlations of the impulse response $h$ and the data $g$ in the output plane $P_3$, the distance $d_1$ which is from the horizontal axis $x_1$ to the top edge of the input plane $P_1$ shown in Figure 2.12 must be bigger than or equal to the distance $D$ obtained in Equation 2.66 which is from the horizontal axis $x_3$ in the output plane $P_3$ to the top edge of the pattern centered at $(0, Y)$ or the bottom edge of the pattern centered at $(0, -Y)$ shown in Figure 2.13; as well as the distance $d_2$ which is from the horizontal axis $x_1$ to the bottom edge of the input plane $P_1$ shown in Figure 2.12 must be bigger than or equal to the distance $D$. For making $d_1$ is equal to $D$, the amount $D - r_1$ has to be added to the distance $r_1$; where the distance $r_1$ obtained in Equation 2.67 is from the horizontal axis $x_1$ to the top edge of $h$ as shown
in Figure 2.12. Similarly, to make \( d_2 \) is equal to \( D \), the amount \( D - r_2 \) has to be added to the distance \( r_2 \); where the distance \( r_2 \) obtained in Equation 2.68 is from the horizontal axis \( x_1 \) to the bottom edge of \( g \) as shown in Figure 2.12. The distances \( D - r_1 \) and \( D - r_2 \) are respectively obtained as in Equation 2.69 and Equation 2.70. Finally, the distance \( d_1 \) will be equal to the distance \( D \) as in Equation 2.71; similarly, the distance \( d_2 \) will also be equal to \( D \) as in Equation 2.72.

\[
D = Y + c + \frac{W_g}{2} + \frac{W_h}{2} = \max\{W_h, W_g\} + W_h + W_g + c
\]  
(2.66)

\[
r_1 = \frac{Y + c + W_g}{2} + \frac{W_h}{2} = \left[ \frac{\max\{W_h, W_g\}}{2} + \frac{W_h}{4} + \frac{W_g}{4} + \frac{c}{2} \right] + \frac{W_h}{2} = \frac{\max\{W_h, W_g\}}{2} + \frac{3}{4}W_h + \frac{1}{4}W_g + \frac{c}{2}
\]  
(2.67)

\[
r_2 = \frac{Y + c + W_g}{2} + \frac{W_h}{2} = \left[ \frac{\max\{W_h, W_g\}}{2} + \frac{W_h}{4} + \frac{W_g}{4} + \frac{c}{2} \right] + \frac{W_g}{2} = \frac{\max\{W_h, W_g\}}{2} + \frac{1}{4}W_h + \frac{3}{4}W_g + \frac{c}{2}
\]  
(2.68)

\[
D - r_1 = \frac{1}{2} \max\{W_h, W_g\} + \frac{1}{4}W_h + \frac{3}{4}W_g + \frac{c}{2}
\]  
(2.69)

\[
D - r_2 = \frac{1}{2} \max\{W_h, W_g\} + \frac{3}{4}W_h + \frac{1}{4}W_g + \frac{c}{2}
\]  
(2.70)

\[
d_1 = r_1 + (D - r_1) = D
\]  
(2.71)

\[
d_2 = r_2 + (D - r_2) = D
\]  
(2.72)

Lastly, the joint transform correlator has a great feature is that its ability to change the filter impulse response quickly; therefore, it is considered beneficial for real-time systems. On the other hand, it has a defect is that the input bandwidth of the data is reduced due to the filter impulse response is introduced simultaneously with the data to be filtered.

2.3.4.1  Sampling Issues

2.3.4.1.1  Introduction

Sampling is considered the first important step for simulating optical models. In order to generate the cross-correlated field in the back focal plane \( P_3 \); then the input plane \( P_1 \), the back focal plane of the lens \( L_2 \) and the back focal plane of the lens \( L_4 \) must be sampled properly. Sampling for each one of these is completely discussed in this sub-subsection.

2.3.4.1.2  Sampling of the Input Plane \( P_1 \)

The input plane \( P_1 \) is located in a rectangular array has \( N \) samples along the spatial space coordinate \( x_1 \) and \( M \) samples along the spatial space coordinate \( y_1 \). \( L_{x_1} \) is the physical side length in meter (\( m \)) of the array in the \( x_1 \) direction; similarly, \( L_{y_1} \) is
the physical side length in meter of the array in the $y_1$ direction. Then the sample spacing $\Delta x_1$ along the $x_1$-coordinate in meter is equal to $\frac{L_{x_1}}{N}$; and the sample spacing $\Delta y_1$ along the $y_1$-coordinate in meter is equal to $\frac{L_{y_1}}{M}$.

2.3.4.1.3 Sampling of the Back Focal Plane $P_2$

From Fourier transform, the spatial frequency coordinate $f_{X_2}$ of the back focal plane $P_2$ in cycles per meter ($\text{cyc m}^{-1}$) is obtained as in Equation 2.73.

$$f_{X_2} = \frac{x_2}{\lambda f}$$  \hspace{1cm} (2.73)

By turning around Equation 2.73, the spatial space coordinate $x_2$ can be expressed as in Equation 2.74.

$$x_2 = \lambda f f_{X_2}$$  \hspace{1cm} (2.74)

Note that, $x_2$ is in meter ($m$) because the units of the variables in Equation 2.74 can be concluded as $m \times m \times \frac{\text{cyc}}{m} = m$. Then the spatial space sampling interval $\Delta x_2$ in
Figure 2.13: The alignment of the patterns of the cross-correlated field in the output plane $P_3$.

Meter ($m$) along the spatial space coordinate $x_2$ is computed as in Equation 2.75.

$$
\Delta x_2 = \lambda f \Delta f x_2
$$

From discrete Fourier transform (DFT), the relationship between the spatial space sampling interval $\Delta x_1$ in the input plane and the spatial frequency sampling interval $\Delta f x_2$ is shown in Equation 2.76.

$$
\Delta f x_2 = \frac{1}{N \Delta x_1}
$$

For more information about how the relation in Equation 2.76 is obtained; Reference [14], Subsection 4.4.2 can be consulted. By substitution from Equation 2.76 in Equation 2.75 then the spatial space sampling interval $\Delta x_2$ in meter ($m$) along the spatial space coordinate $x_2$ is obtained as in Equation 2.77.

$$
\Delta x_2 = \frac{\lambda f}{N \Delta x_1}
$$

Similarly, the spatial space sampling interval $\Delta y_2$ in meter ($m$) along the spatial
space coordinate $y_2$ can be computed as in Equation 2.78.

$$\Delta y_2 = \frac{\lambda f}{M \Delta y_1} \quad (2.78)$$

### 2.3.4.1.4 Sampling of the Back Focal Plane $P_3$

From Fourier transform, the spatial frequency coordinate $f_{X_3}$ of the back focal plane $P_3$ in cycles per meter ($\text{cyc/m}$) is obtained as in Equation 2.79.

$$f_{X_3} = \frac{x_3}{\lambda f} \quad (2.79)$$

By turning around Equation 2.79, the spatial space coordinate $x_3$ can be expressed as in Equation 2.80.

$$x_3 = \lambda f f_{X_3} \quad (2.80)$$

Then the spatial space sampling interval $\Delta x_3$ in meter ($m$) along the spatial space coordinate $x_3$ is computed as in Equation 2.81.

$$\Delta x_3 = \lambda f \Delta f_{X_3} \quad (2.81)$$

From discrete Fourier transform (DFT), the relationship between the spatial space sampling interval $\Delta x_2$ in the back focal plane $P_2$ and the spatial frequency sampling interval $\Delta f_{X_3}$ is shown in Equation 2.82.

$$\Delta f_{X_3} = \frac{1}{N \Delta x_2} \quad (2.82)$$

By substitution from Equation 2.82 in Equation 2.81 then the spatial space sampling interval $\Delta x_3$ in meter ($m$) along the spatial space coordinate $x_3$ is obtained as in Equation 2.83.

$$\Delta x_3 = \lambda f \frac{1}{N \Delta x_2} = \Delta x_1 \quad (2.83)$$

Similarly, the spatial space sampling interval $\Delta y_3$ in meter ($m$) along the spatial space coordinate $y_3$ can be computed as in Equation 2.84.

$$\Delta y_3 = \frac{\lambda f}{M \Delta y_2} = \Delta y_1 \quad (2.84)$$

The spatial space sampling intervals in the input plane $P_1$ and the back focal plane $P_3$ are equal because the focal lengths of the lenses $L_2$ and $L_4$ are equal; but if the focal lengths are not identical then the sampling intervals will not be equal anymore.
2.3.4.2 Applications of the JTC

2.3.4.2.1 Introduction

The joint transform correlator (JTC) is one of the techniques frequently used in the field of optical pattern identification and classification. It plays an important role in object detection, face recognition, fingerprint recognition, and many other areas. In this sub-subsection, the usage of the JTC for object detection and face recognition is fully covered.

2.3.4.2.2 Face Recognition

The joint transform correlator (JTC) is used to recognize an unknown face object based on a database of desired impulses. The database is composed of \( n, N \times N \) images of faces (impulses) on black backgrounds such that \( I_k \) where \( k = 1, \ldots, n \); and \( n \) is the total number of the impulses. For decreasing the error rates of face recognition and object detection, all face projections must be defined in the database; impulses must have the same size; impulses must have only faces and they are expanded to the boarders of the images; finally, impulses must have unified backgrounds to discriminate between the pixels occupying the backgrounds and the pixels on the faces. For doing face recognition, an unknown face object have to be picked. It must have the same properties of the impulses. Hence, it must have the same size as the impulses; it has a black background; it has a projection as one of the impulses; and it has only face that is expanded to the boarders of the image. The pixels occupying the faces of the impulses and the unknown face object are normalized in the same manner as the normalization of the pixels occupying the faces of the training faces in Step 2 in Sub-subsection 2.2.2.6.

The cross-correlated field between the unknown face object and each impulse is obtained. An adaptive filtering mask is used to produce the cross-correlated patterns that are centered at \((0, Y)\). The maximum values of the cross-correlated patterns are computed; such that \( c_1 \) is the maximum value of the cross-correlated pattern between the unknown face object and the first impulse; similarly, \( c_n \) is the maximum value of the cross-correlated pattern between the unknown face object and the \( n^{th} \) impulse.

Then recognition can be performed by using Condition 2.4 where \( b_{c_k} \) is the biggest cross-correlation among the maximum values of the cross-correlated patterns between the unknown face object and impulses.

Condition 2.4. If,

\[
b_{c_k} = \max ([c_1, \ldots, c_n]), \quad \text{where } k \text{ can be any number between } 1 \text{ to } n
\]

Then the unknown face object is recognized as the \( k^{th} \) impulse response; otherwise, it is an unknown face object.
2.3.4.2.2.1 Setting up a Recognition Threshold

Unfortunately, when the unknown face object does not have a similar impulse response then getting the biggest cross-correlation $bc_k$ among the maximum values of the cross-correlated patterns does not always mean that the unknown face object is recognized as the $k^{th}$ impulse response. Therefore, a certain threshold must be used to increase the accuracy of recognition.

For setting up a recognition threshold, some different objects are taken for each impulse; consequently, the objects are known here just for picking the threshold. The cross-correlated patterns that are centered at $(0, Y)$ resulting from the cross-correlations between each impulse and its corresponding objects are produced. After that, the maximum values of the cross-correlated patterns between the impulses and their corresponding objects are obtained; and they are stacked in the row vectors $T_1, \ldots, T_k, \ldots, T_n$, where $T_k$ contains the biggest expected cross-correlations because the $k^{th}$ impulse response and its corresponding objects have the same person and the same face position.

Thereafter, the mean $m_k$ and the standard deviation $STD_k$ (the average distance from the mean to a point) are computed for each row vector $T_k$ of the biggest cross-correlations. By calculating the means and the standard deviations, a certain threshold is established for each impulse response.

Then the recognition threshold can be applied to recognize the unknown face object by using Condition 2.5.

**Condition 2.5.** If,

$$m_k - STD_k \leq bc_k \leq m_k + STD_k,$$

where $k = 1, \ldots, n$,

then the unknown face object is recognized as the $k^{th}$ impulse response; otherwise, it is an unknown face object.

2.3.4.2.3 Object Detection

The joint transform correlator (JTC) is used to detect if an unknown object contains a face or not based on a determined threshold for detection. Therefore, in object detection, only a detection threshold is needed.

To obtain the detection threshold, objects are generated from the database of the impulses in the same manner as in Sub-sub-sub-subsection 2.3.4.2.2.1. The cross-correlated patterns that are centered at $(0, Y)$ resulting from the cross-correlations between each impulse and all objects are produced. The maximum values of the cross-correlated patterns between the impulses and all objects are computed and placed in the row vector $S$. After that, the mean $m_S$ and the standard deviation $STD_S$ (the average distance from the mean to a point) are computed for the vector $S$. By calculating the mean and the standard deviation, the detection threshold is established.
For applying the detection threshold, the unknown object $I^{unknown}$ is picked for detection. The maximum value $c^{unknown}$ of the cross-correlated pattern between the unknown object and any impulse response is computed. Then the unknown object $I^{unknown}$ can be detected by means of Condition 2.6.

**Condition 2.6.** If,

$$m_S - STD_S \leq c^{unknown} \leq m_S + STD_S$$

Then the unknown object $I^{unknown}$ is detected as a face object; otherwise, it is not a face object.

### 2.3.4.3 Analysis of JTC Performance

#### 2.3.4.3.1 Introduction

Joint transform correlator (JTC) performance is analyzed through analyzing the performance of its applications. In order to study the performance of face recognition and object detection, their error rates are computed and analyzed. Hence, in this sub-subsection, attention is paid for calculating the error rates of face recognition and object detection.

#### 2.3.4.3.2 Analysis of Recognition Performance

The error rate computes the percentage of error in recognition. For computing the error rate of face recognition, let $L$ is the total number of objects; $SR$ is the total number of successes in the recognition of the objects; and $FR$ is the total number of failures in the recognition of the objects. Then the error rate can be computed as in Equation 2.85.

$$ER(\%) = \frac{FR}{L} \times 100 \quad (2.85)$$

#### 2.3.4.3.2.1 Improvement of Recognition Performance

The error rate of face recognition is usually big then it has to be improved. One of the techniques for decreasing the error rate is optimizing the database of the impulses. The optimization of the database means trying to find the best combination of impulses that ensures the smallest error rate.

Some different objects are taken for each impulse response; consequently, the objects are known here just for optimizing the database of the impulses. Then this technique can be simulated by trying different combinations out of the generated objects until finding a combination that produces the lowest error rate of face recognition then the combination can be used to form the database of the impulses.
2.3.4.3.3 Analysis of Detection Performance

The error rate computes the percentage of error in detection. For computing the error rate of object detection, let \( L \) is the total number of objects; \( SD \) is the total number of successes in the detection of the objects; and \( FD \) is the total number of failures in the detection of the objects. Then the error rate can be computed as in Equation 2.86.

\[
ER(\%) = \frac{FD}{L} \times 100
\]  

(2.86)

2.3.4.3.3.1 Improvement of Detection Performance

The error rate of object detection is usually big then it has to be improved. The error rate is improved by optimizing the database of the impulses in the same manner as the optimization of the error rate of face recognition in Sub-sub-sub-subsection 2.3.4.3.2.1.
Chapter 3

Modelling

3.1 Digital Modelling

3.1.1 Introduction

In this section, PCA and IPCA algorithms are simulated by using the MATLAB © software when the data set is composed of images; as well as comparison between two algorithms is illustrated. The simulation begins from generating the database of the training faces until projecting the training faces on the eigenspace. The applications of the PCA and IPCA algorithms are simulated too. The complete MATLAB © code that had been written to simulate this work is shown in Appendix A.

3.1.2 Application of the PCA Algorithm to Images

The steps for analyzing the PCA algorithm to images presented in Sub-subsection 2.2.2.6 are simulated as follows:

Step 1: the database is created to be composed of nine, $50 \times 50$ images of faces (training faces) on black backgrounds such that $I_k$ where $k = 1, \ldots, 9$. The training faces are taken for three people where each person has three training faces with different projections as shown in Figure 3.1.

Step 2: the average histogram for all training faces is obtained as in Figure 3.2. From Figure 3.2, it turns out all intensity levels below the threshold represent the backgrounds of images because these levels have the biggest histogram; and all intensity levels above the threshold represent the faces; therefore, the threshold is equal to eight. The threshold is applied on the training faces; Figure 3.3 shows how good the applied threshold is on the training faces. As seen from Figure 3.3, the applied
threshold is doing pretty well; and it can be used for normalizing the training faces as in Figure 3.4.

Figure 3.1: The database of the training faces.

Figure 3.2: The average histogram for all training faces.
Figure 3.3: The application of the threshold on the training faces.
Figure 3.4: The normalization of all training faces by means of the selected threshold.
Chapter 3. Modelling

Step 3: all training faces are centered as shown in Figure 3.5.

**Figure 3.5:** The centered training faces.

Step 4: all centered training faces are represented as $50^2$ column vectors.

Step 5: The average training face vector $\Psi$ is computed and the average training face is shown in Figure 3.6.

**Figure 3.6:** The average training face. Note that, this is a negative image.
Step 6: the set of the training faces is centered; and the centered (with respect to the set of the training faces) training faces are presented in Figure 3.7.

![Figure 3.7](image)

**Figure 3.7:** The centered (with respect to the set of the training faces) training faces. Note that, these are negative images.

Step 7: a $2500 \times 2500$ covariance matrix for all training faces is calculated.
Step 8: the eigenvectors and eigenvalues of the covariance matrix $C$ are computed.

Step 9: for choosing principal components and forming the feature vector matrix $A_q$, the variance contribution rate ($VCR$) and the total variance contribution rate ($TVC$) are computed in Table 3.1. When the $TVC$ is over 95.5% then $q$ eigenvectors associated with the biggest $q$ eigenvalues are selected. Hence, Based on Table 3.1, the biggest eight eigenvectors associated with the biggest eigenvalues are selected to form the feature vector matrix. The selected eigenfaces (the highly correlated eigenvectors) are shown in Figure 3.8.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$\lambda_k$</th>
<th>$VCR_k$ (%)</th>
<th>$TVC$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9732585.1016</td>
<td>29.0361</td>
<td>29.0361</td>
</tr>
<tr>
<td>2</td>
<td>6578765.9806</td>
<td>19.6270</td>
<td>48.6631</td>
</tr>
<tr>
<td>3</td>
<td>4576184.9630</td>
<td>13.6525</td>
<td>62.3157</td>
</tr>
<tr>
<td>4</td>
<td>4054857.8169</td>
<td>12.0972</td>
<td>74.4129</td>
</tr>
<tr>
<td>5</td>
<td>3000248.2414</td>
<td>8.9509</td>
<td>83.3638</td>
</tr>
<tr>
<td>6</td>
<td>2204809.6081</td>
<td>6.5778</td>
<td>89.9416</td>
</tr>
<tr>
<td>7</td>
<td>1816000.6101</td>
<td>5.4178</td>
<td>95.3595</td>
</tr>
<tr>
<td>8</td>
<td>1555452.9118</td>
<td>4.6405</td>
<td>100</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

Step 10: the principal components transform is performed and the training faces are projected on the eigenspace.

### 3.1.3 Application of the IPCA Algorithm to Images

The IPCA algorithm shown in Sub-subsection 2.2.3.2 is used for calculating the $IR$ and $AIR$ for the computed eigenvectors of the covariance matrix $C$ in Subsection 3.1.2 where $d$ here is equal to the total number of the training faces $n$; and the results are shown in Table 3.2. When the $AIR$ is over 95.5% then $q$ eigenvectors associated with the biggest $q$ eigenvalues are selected. Hence, Based on Table 3.2, the biggest seven eigenvectors associated with the biggest eigenvalues are selected to form the feature vector matrix $A_q$. The selected eigenfaces (the highly correlated eigenvectors) are the first seven selected eigenfaces by using the PCA algorithm shown in Figure 3.8.
3.1.4 Comparison of the PCA and IPCA Algorithms

Comparison between the PCA and IPCA algorithms is based on the values of the $TVC$ and $AIR$ that determine the selected eigenfaces for the feature vector matrix $A_q$ in Subsection 3.1.2, Step 9 and in Subsection 3.1.3.

When $k = 7$ in Table 3.2, the $AIR$ is equal to 95.6891% which is bigger than 95.5% but the $TVC$ is slightly small; consequently, if the eigenvectors are selected

Figure 3.8: The eigenfaces. Note that, these are negative images.
Table 3.2: The calculations of the $IR$ and $AIR$.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$\lambda_k$</th>
<th>$\rho_k$</th>
<th>$I (\lambda_k)$</th>
<th>$IR_k$ (%)</th>
<th>$AIR$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9732585.1016</td>
<td>0.7096</td>
<td>0.4948</td>
<td>31.1180</td>
<td>31.1180</td>
</tr>
<tr>
<td>2</td>
<td>6578765.9806</td>
<td>0.8037</td>
<td>0.3152</td>
<td>19.8224</td>
<td>50.9404</td>
</tr>
<tr>
<td>3</td>
<td>4576184.9630</td>
<td>0.8635</td>
<td>0.2118</td>
<td>13.3174</td>
<td>64.2578</td>
</tr>
<tr>
<td>4</td>
<td>4054857.8169</td>
<td>0.8790</td>
<td>0.1860</td>
<td>11.6978</td>
<td>75.9555</td>
</tr>
<tr>
<td>5</td>
<td>3000248.2414</td>
<td>0.9105</td>
<td>0.1353</td>
<td>8.5073</td>
<td>84.4628</td>
</tr>
<tr>
<td>6</td>
<td>2204809.6081</td>
<td>0.9342</td>
<td>0.0982</td>
<td>6.1729</td>
<td>90.6357</td>
</tr>
<tr>
<td>7</td>
<td>1816000.6101</td>
<td>0.9458</td>
<td>0.0804</td>
<td>5.0534</td>
<td>95.6891</td>
</tr>
<tr>
<td>8</td>
<td>1555452.9118</td>
<td>0.9536</td>
<td>0.0686</td>
<td>4.3109</td>
<td>100</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Based on the $TVC$ then the first eight eigenvectors must be taken in order to make sure that the $TVC$ is big enough. Therefore, the $AIR$ tells us more about the information contained in the eigenfaces. Figure 3.9 shows a comparison between the biggest calculated eigenvalues by using the PCA and IPCA algorithms as well as the calculated eigenvalues from the covariance matrix $C$ for all training faces.

Figure 3.9: A comparison between the biggest calculated eigenvalues by using the PCA and IPCA algorithms as well as the calculated eigenvalues from the covariance matrix for all training faces.
3.1.5 Image Compression

In image compression modelling, the training faces in Figure 3.1 are projected on the eigenspace and reconstructed again by using the PCA and IPCA algorithms; as well as by using different selected eigenfaces to form the feature vector matrix $A_q$.

3.1.6 Face Recognition

The database of the training faces created in Step 1 in Subsection 3.1.2 is used for recognition. Some tested images are taken out of the database for setting up a recognition threshold. Thirty-six tested images are taken for the first projection of each person; twelve tested images are taken for the second projection of each person; and twelve tested images are taken for third projection of each person; therefore, the total number of the tested images is equal to 180. Some samples of the tested images are shown in Figure 3.10; the full database of the tested images is shown in Appendix B.

By using the PCA algorithm, a recognition threshold is specified for each training face by means of the method explained in Sub-sub-subsection 2.2.5.3.1. The tested images are processed as unknown images of faces as illustrated in Sub-subsection 2.2.5.3. Then Condition 2.2 is used for recognizing the tested images. The recognition results of the tested images in Figure 3.10 by using the PCA algorithm are presented in Table 3.3. The recognition of all 180 tested images by using the PCA algorithm is shown in Appendix C. Finally, the tested images can be recognized in the same way when the IPCA algorithm is used; or different eigenfaces are selected to form the feature vector matrix.

<table>
<thead>
<tr>
<th>Tested Image No.</th>
<th>Input Face</th>
<th>Recognized Output Face</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>15</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>38</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>62</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>72</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>120</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>124</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>160</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>179</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
</tbody>
</table>
3.1.7 Image Detection

The generated database of the tested images in Subsection 3.1.6 is used for detection. By using the PCA algorithm, a detection threshold is specified by means of the method explained in Sub-subsection 2.2.5.4. Then Condition 2.3 is used for detecting
the tested images. The detection results of the tested images in Figure 3.10 by using the PCA algorithm are presented in Table 3.4. The detection of all 180 tested images by using the PCA algorithm is shown in Appendix D. Finally, the tested images can be detected in the same way when the IPCA algorithm is used; or different eigenfaces are selected to form the feature vector matrix.

Table 3.4: The detection of the tested images in Figure 3.10.

<table>
<thead>
<tr>
<th>Tested Image No.</th>
<th>Input Image</th>
<th>Detected Output Image</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>15</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>38</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>62</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>72</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>120</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>124</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>160</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>179</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
</tbody>
</table>

3.2 Optical Modelling

3.2.1 Introduction

In this section, the joint transform correlator (JTC) is fully simulated by using the MATLAB software. The simulation begins from scratch until generating the desired pattern of the cross-correlated field in the back focal plane $P_3$ by using an adaptive filtering mask designed for that. The JTC applications are simulated too. The complete MATLAB code that had been written to simulate this work is shown in Appendix F.

In the simulation, lenses are assumed to be ideally focused (i.e. the model is an aberration-free system) as in Figure 3.11; where $f$ in the figure is the focal length.

3.2.2 Simulation of the JTC

We are at level that we can start in simulating the joint transform correlator (JTC). The desired impulse response $h$ and the data $g$ (here it is called the object) are $100 \times 100$ images of faces for two people. They are respectively shown in Figure 3.12 and Figure 3.13. The impulse response and the object are normalized in order to remove lighting effects on them then increasing the accuracy of cross-correlation. To
keep variations among them just in the faces without the effects of the backgrounds, the normalization is performed on the pixels occupying the faces in the same manner as the normalization of the pixels on the faces of the training faces in Step 2 in Sub-subsection 2.2.2.6.

The transparencies of the impulse response $h$ and the object $g$ are located in a square array in the input plane $P_1$. The width $W_g$ of $g$ in the direction of the $y$-coordinate is 100 pixels; the width $W_h$ of $h$ in the direction of the $y$-coordinate is 100
pixels; then the distance $Y$ that separates the centers of $h$ and $g$ is equal to 200 pixels. The constant $c$ is selected to be 10 pixels then Relation 2.1 is satisfied. The distance $D$ obtained in Equation 2.66 is 130 pixels; the distance $r_1$ obtained in Equation 2.67 is 155 pixels; the distance $r_2$ obtained in Equation 2.68 is 155 pixels; $D - r_1$ and $D - r_2$ are respectively equal to 155 pixels and 155 pixels; then the distances $d_1$ and $d_2$ obtained respectively in Equation 2.71 and Equation 2.72 are respectively equal to 310 pixels and 310 pixels. Since the distance $d_1$ is equal to the distance $D$ as well as the distance $d_2$ is also equal to the distance $D$ then the input transparencies of $h$ and $g$ are aligned properly in the input plane $P_1$.

The number of samples $N$ along the spatial space coordinate $x_1$ in the input plane $P_1$ is 630; and the number of samples $M$ along the spatial space coordinate $y_1$ is 630. The physical side length $L_{x_1}$ of the array in the $x_1$ direction is $10 \, (m)$; and the physical side length $L_{y_1}$ of the array in the $y_1$ direction is $10 \, (m)$. Then the sample spacing $\Delta x_1$ along the $x_1$-coordinate is equal to $\frac{10}{630} = 0.0159 \, (m)$; and the sample spacing $\Delta y_1$ along the $y_1$-coordinate is equal to $\frac{10}{630} = 0.0159 \, (m)$. The transmitted field $U_1(x_1, y_1)$ from the input plane $P_1$ is shown in Figure 3.14.

![The Transmitted Field from the Input Plane $P_1$](image)

**Figure 3.14:** The transmitted field $U_1(x_1, y_1)$ from the input plane $P_1$.

The light wavelength $\lambda$ is $550 \times 10^{-9} \, (m)$; and the focal length $f$ is $0.055 \, (m)$. Then the spatial space sampling interval $\Delta x_2$ along the spatial space coordinate $x_2$ is equal to $\frac{M}{N} \Delta x_1 = \frac{550 \times 10^{-9} \times 0.055}{630 \times 0.0159} = 3.0250 \times 10^{-9} \, (m)$; similarly, the spatial space sampling
interval $\Delta y_2$ along the spatial space coordinate $y_2$ is equal to $\frac{\lambda f}{M \Delta y_1} = \frac{550 \times 10^{-9} \times 0.055}{630 \times 0.0159} = 3.0250 \times 10^{-9} (m)$. The incident intensity $I(x_2, y_2)$ on the back focal plane $P_2$ is shown in Figure 3.15.

![The incident intensity on the plane $P_2$.](image)

**Figure 3.15:** The incident intensity $I(x_2, y_2)$ on the back focal plane $P_2$.

Since the focal lengths of the lenses $L_2$ and $L_4$ are equal, the spatial space sampling intervals $\Delta x_3$ and $\Delta y_3$ in the back focal plane $P_3$ are respectively equal to the spatial space sampling intervals $\Delta x_1$ and $\Delta y_1$ in the input plane $P_1$. The cross-correlated field $U_3(x_3, y_3)$ in the back focal plane $P_3$ is obtained as in Figure 3.16 by calculating the inverse Fourier transform for the incident intensity $I(x_2, y_2)$ on the back focal plane $P_2$.

Figure 3.17 shows the designed adaptive mask for obtaining the desired pattern of the cross-correlations of the impulse response $h$ and the object $g$ in the back focal plane $P_3$. The mask produces the cross-correlated pattern that is centered at $(0, Y)$. It is multiplied by the cross-correlated field in the plane $P_3$ in order to obtain the filtered field as in Figure 3.18.
Figure 3.16: The cross-correlated field $U_3(x_3, y_3)$ in the back focal plane $P_3$.

Figure 3.17: The adaptive filtering mask.
3.2.3 Face Recognition

The database of the impulses for face recognition is picked to contain three, 100 × 100 impulses (images of faces) on black backgrounds. The impulses are taken for vertical faces to people’s shoulders where oblique faces are ignored as shown in Figure 3.19. Taking just vertical faces will simplify the optimization of the database of the impulses. Some objects are taken out of the database for setting up a recognition threshold. Thirty-six objects are taken for each impulse then the total number of the objects is equal to 108. Some samples of the objects are shown in Figure 3.20; the full database of the objects is shown in Appendix A.

A recognition threshold is specified for each impulse by means of the method explained in Sub-sub-sub-subsection 2.3.4.2.2.1. The objects are processed as unknown images of faces as illustrated in Sub-sub-subsection 2.3.4.2.2. Then Condition 2.5 is used for recognizing the objects. The recognition results of the objects in Figure 3.20 are presented in Table 3.5. The recognition of all 108 objects is shown in Appendix G.
Section 3.2. Optical Modelling

Figure 3.19: The database of the impulses.

Figure 3.20: Some samples of the objects.

Table 3.5: The recognition of the objects in Figure 3.20.

<table>
<thead>
<tr>
<th>Object No.</th>
<th>Input Face</th>
<th>Recognized Output Face</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>8</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mohammed Hanafy</td>
<td>Failure</td>
</tr>
<tr>
<td>40</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>72</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>76</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>83</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
</tbody>
</table>
3.2.4 Object Detection

The database of the generated objects in Subsection 3.2.3 is used for detection. A detection threshold is specified by means of the method explained in Sub-sub-subsection 2.3.4.2.3. Then Condition 2.6 is used for detecting the objects. The detection results of the objects in Figure 3.20 are presented in Table 3.6. The detection of all 108 objects is shown in Appendix H.

Table 3.6: The detection of the objects in Figure 3.20.

<table>
<thead>
<tr>
<th>Object No.</th>
<th>Input Object</th>
<th>Detected Output Object</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>8</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>40</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>72</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>76</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>83</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
</tbody>
</table>
Chapter 4. Results of Performance Analysis

Results of Performance Analysis

4.1 Performance Results of the PCA and IPCA Algorithms

4.1.1 Introduction

After modelling the applications of the PCA and IPCA algorithms, the performance results for each application are obtained in this subsection. The results show the behavior of each application when the eigenfaces that are generated by using the PCA algorithm are used; the eigenfaces that are generated by using the IPCA algorithm are used; and when different eigenfaces are selected to form the feature vector matrix.

4.1.2 Results of Compression Performance

4.1.2.1 Speed of Compression and Reconstruction

When a small number of the eigenfaces is used to project and reconstruct the training faces then the processing speed will increase and vice versa. Therefore, the IPCA algorithm is the fastest one; then the PCA algorithm comes second; finally, the smallest processing speed occurs when all calculated eigenvectors from the covariance matrix for all training faces are used as eigenfaces.

4.1.2.2 Quality of a Reconstructed Image

When a small number of the eigenfaces is used to project and reconstruct the training faces then the training faces will have bad quality. Therefore, the highest error in reconstruction occurs when the IPCA algorithm is used; then the PCA algorithm
comes second; finally, the usage of all eigenvectors as eigenfaces produces the smallest reconstruction error.

For measuring the quality of the reconstructed training faces, Equation 2.46 is used for computing the mean squared errors (MSEs) between the training faces and their reconstructions. Figure 4.1 shows the plot of the MSEs of reconstructing training face number five for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used. The plots of the MSEs of reconstructing other training faces are shown in Appendix I. Figure 4.2 shows the reconstruction of training face number five by using the highest correlated eigenface ($q = 1$), the PCA eigenfaces ($q = 8$), the IPCA eigenfaces ($q = 7$), and all eigenvectors as eigenfaces ($q = 2500$); along with mean squared errors resulted from reconstructing the training face by using those eigenfaces.

**Figure 4.1:** The plot of the mean squared errors (MSEs) of reconstructing training face number five for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used.
Figure 4.2: The reconstruction of training face number five by using the highest correlated eigenface \((q = 1)\), the PCA eigenfaces \((q = 8)\), the IPCA eigenfaces \((q = 7)\), and all eigenvectors as eigenfaces \((q = 2500)\); along with mean squared errors resulted from reconstructing the training face by using those eigenfaces.

From Figure 4.1 and Figure 4.2, it can be noticed that when all eigenvectors
are used as eigenfaces then the reconstructed training faces have the worst resolution; that because the covariance matrix is too big then the calculation of 2500 eigenvectors by using the MATLAB © software leads to some round-off errors in the eigenvectors associated with the smallest eigenvalues. Hence, it is not recommended to select the lowest correlated eigenvectors for reconstructing training faces due to they add some noise to the reconstructed training faces. In addition, due to round-off error, the MATLAB © software makes the smallest eigenvalues negative while they must be positive because they are calculated from a positive definite covariance matrix. Those negative eigenvectors must be set to zero.

4.1.2.3 Size of Compression

4.1.2.3.1 Information Rate

Obviously, from Equation 2.47, an information rate depends on the number of the selected eigenfaces \( q \). Note that, when all eigenvectors are picked to form the feature vector matrix then there will not be any compression; and the overall size when there is no any compression method is used will be the optimum one. Consequently, when the number of the selected eigenfaces decreases then an information rate will decrease (i.e. compression will increase) and vice versa. Therefore, the IPCA algorithm offers the highest compression with the highest loss of information; after that, when the PCA algorithm is used, there will not be information lost (i.e. there is no compression); lastly, the usage of all eigenvectors as eigenfaces add some information (i.e. there is no compression).

For measuring how much information is after compression compared with information before compression, Equation 2.47 is used. Figure 4.3 shows the rates of information for different selected eigenfaces compared with resulted information rates when the PCA and IPCA algorithms are used.

4.1.2.3.2 Mean Squared Error (MSE) of Compressed Images

Lost information increases when a small number of eigenfaces is picked and vice versa. Therefore, the IPCA algorithm offers the highest compression with the highest error; then the PCA algorithm comes second; lastly, the usage of all eigenvectors as eigenfaces produce the lowest compression with the lowest error.

Equation 2.54 is used for computing the mean squared error (MSE) of compression. Figure 4.4 shows the mean squared errors (MSEs) of compression for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used.
4.1.3 Results of Recognition Performance

4.1.3.1 Speed of Recognition

When the number of the selected eigenfaces decreases, the processing speed increases and vice versa. Therefore, the usage of all calculated eigenvectors as eigenfaces leads to the biggest processing time; then the usage of the calculated eigenfaces by using the PCA algorithm comes second; finally, the usage of the calculated eigenfaces by using the IPCA algorithm leads to the smallest processing time.

4.1.3.2 Error Rate

When the number of the selected eigenfaces increases, the error rate decreases; that because the unknown face image will be projected precisely next to its corresponding training face. On the other hand, when the selected eigenfaces decreases, the error rate increases. Therefore, the usage of all calculated eigenvectors as eigenfaces leads to the smallest error rate; then the usage of the calculated eigenfaces by using the PCA algorithm comes second; finally, the usage of the calculated eigenfaces by using
Section 4.1. Performance Results of the PCA and IPCA Algorithms

Figure 4.4: The mean squared errors (MSEs) of compression for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used.

the IPCA algorithm leads to the biggest error rate.

In the recognition of all 180 tested images in Subsection 3.1.6, \( L = 180; SR = 133; \) and \( FR = 47. \) Then by using Equation 2.55, the error rate \( ER(\%) \) is equal to \( \frac{47}{180} \times 100 = 26.1111\% . \) Figure 4.5 shows the error rates of recognition for different selected eigenfaces compared with resulted error rates when the PCA and IPCA algorithms are used.

4.1.4 Results of Detection Performance

4.1.4.1 Speed of Detection

When the number of the selected eigenfaces decreases, the processing speed increases and vice versa. Therefore, the IPCA algorithm is the fastest one; then the PCA algorithm comes second; finally, the smallest processing speed occurs when all calculated eigenvectors from the covariance matrix for all training faces are used as eigenfaces.
4.1.4.2 Error Rate

When the eigenfaces that contain the most significant patterns from the correlated training faces (highly correlated eigenvectors) are only selected then the accuracy of face detection increases. That because in face detection, distance calculation is between the centered unknown image and its reconstruction; consequently, if the unknown image is not a face then the distance will be big; therefore, the unknown image will not be detected as a face image. As a result of that, the usage of the calculated eigenfaces by using the IPCA algorithm produces the smallest error rate; then the usage of the calculated eigenfaces by using the PCA algorithm comes second; finally, the usage of all calculated eigenvectors as eigenfaces obtains the biggest error rate.

In the detection of all 180 tested images in Subsection 3.1.7, \( L = 180; \ SD = 147; \) and \( FD = 33. \) Then by using Equation 2.56, the error rate \( ER(\%) \) is equal to \( \frac{33}{180} \times 100 = 18.3333\% . \) Figure 4.6 shows the error rates of detection for different selected eigenfaces compared with resulted error rates when the PCA and IPCA algorithms are used.
4.2 Results of JTC Performance

4.2.1 Results of Recognition Performance

In the recognition of all 108 objects in Subsection 3.2.3, \( L = 108 \); \( SR = 20 \); and \( FR = 88 \). Then by using Equation 2.85, the error rate \( ER(\%) \) is equal to \( \frac{88}{108} \times 100 = 81.4815\% \).

4.2.1.1 Improvement of Recognition Performance

For finding the optimal combination out of the 108 objects in Subsection 3.2.3, 46656 \((36 \times 36 \times 36)\) iterations are performed until the database of the optimal impulses for face recognition is obtained as in Figure 4.7. The complete MATLAB © code that had been written to optimize the database of the impulses for face recognition is shown in Appendix J. When the database of the optimal impulses is used for recognizing all 108 objects then the total number of successes \( SR \) becomes 71;
and the total number of failures $FR$ becomes 37. Therefore, the error rate $ER(\%)$ is equal to $\frac{37}{108} \times 100 = 34.2593\%$ which is much less than the resulted error rate when the database of the impulses is not optimized.

\[ ER(\%) = \frac{37}{108} \times 100 = 34.2593\% \]

Figure 4.7: The database of the optimal impulses for face recognition.

### 4.2.2 Results of Detection Performance

In the detection of all 108 objects in Subsection 3.2.4, $L = 108$; $SD = 58$; and $FD = 50$. Then by using Equation 2.86, the error rate $ER(\%)$ is equal to $\frac{50}{108} \times 100 = 46.2963\%$.

\[ ER(\%) = \frac{50}{108} \times 100 = 46.2963\% \]

#### 4.2.2.1 Improvement of Detection Performance

For finding the optimal combination out of the 108 objects in Subsection 3.2.4, $46656 (36 \times 36 \times 36)$ iterations are performed until the database of the optimal impulses for object detection is obtained as in Figure 4.8. The complete MATLAB ® code that had been written to optimize the database of the impulses for object detection is shown in Appendix J. When the database of the optimal impulses is used for detecting all 108 objects then the total number of successes $SD$ becomes 79; and the total number of failures $FD$ becomes 29. Therefore, the error rate $ER(\%)$ is equal to $\frac{29}{108} \times 100 = 26.8519\%$ which is much less than the resulted error rate when the database of the impulses is not optimized.

\[ ER(\%) = \frac{29}{108} \times 100 = 26.8519\% \]

Figure 4.8: The database of the optimal impulses for object detection.
Conclusion

5.1 Discussion of Results

5.1.1 Introduction

In fact, the results of the joint transform correlator (JTC) applications are obviously shown in Section 4.2; but the results of the PCA and IPCA applications are not summarized yet. Hence, in this section, these results are fully discussed and summarized.

5.1.2 Comparison of the PCA and IPCA Algorithms

5.1.2.1 Introduction

In this sub-subsection, it is determined which algorithm behaves better in each application. It concludes that the IPCA algorithm, in general, behaves better than the PCA algorithm in the most of the applications.

It is very important to be noticed that the calculation of all eigenvectors from the covariance matrix for all training faces is too difficult because the covariance matrix is too big as explained in Step 8 in Sub-subsection 2.2.2.6. Therefore, it is impractical to use all eigenvectors as eigenfaces; but they are computed for comparison purposes with the PCA and IPCA algorithms.

5.1.2.2 Results of Image Compression

The results of image compression are summarized in Table 5.1. From Table 5.1, the IPCA algorithm behaves better than any other algorithm or technique. It offers wonderful compression, reconstruction and processing speed with acceptable errors.
Table 5.1: The results of image compression.

<table>
<thead>
<tr>
<th>The Number of Selected Eigenfaces $q$</th>
<th>The Mean Squared Error (MSE) of Reconstructing Training Face No. 5</th>
<th>An information rate (%) Before Compression : After Compression</th>
<th>The MSE of Compression</th>
<th>Processing Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q = 1$</td>
<td>$2.96 \times 10^6$</td>
<td>100 : 22.26</td>
<td>$2.38 \times 10^5$</td>
<td>The fastest</td>
</tr>
<tr>
<td>$q = 7$ (IPCA Algorithm)</td>
<td>1706.72</td>
<td>100 : 98.17</td>
<td>$1.56 \times 10^6$</td>
<td>Second</td>
</tr>
<tr>
<td>$q = 8$ (PCA Algorithm)</td>
<td>$1.25 \times 10^{-21}$</td>
<td>100 : 100.32</td>
<td>$1.16 \times 10^{-7}$</td>
<td>Third</td>
</tr>
<tr>
<td>$q = 2500$ (All Eigenvectors)</td>
<td>$2.35 \times 10^6$</td>
<td>100 : 27888.9</td>
<td>0</td>
<td>The slowest</td>
</tr>
</tbody>
</table>

5.1.2.3 Results of Face Recognition

The results of face recognition are summarized in Table 5.2. From Table 5.2, the PCA algorithm behaves better than any other algorithm or technique. It offers an acceptable error rate, easy calculation and the speed is not bad.

Table 5.2: The results of face recognition.

<table>
<thead>
<tr>
<th>The Number of Selected Eigenfaces $q$</th>
<th>An Error Rate (%)</th>
<th>Processing Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q = 1$</td>
<td>65.56</td>
<td>The fastest</td>
</tr>
<tr>
<td>$q = 7$ (IPCA Algorithm)</td>
<td>26.67</td>
<td>Second</td>
</tr>
<tr>
<td>$q = 8$ (PCA Algorithm)</td>
<td>26.11</td>
<td>Third</td>
</tr>
<tr>
<td>$q = 2500$ (All Eigenvectors)</td>
<td>17.78</td>
<td>The slowest</td>
</tr>
</tbody>
</table>

5.1.2.4 Results of Face Detection

The results of face detection are summarized in Table 5.3. From Table 5.3, the IPCA algorithm behaves better than any other algorithm or technique. It offers the smallest error rate as well as remarkable speed.
Table 5.3: The results of face detection.

<table>
<thead>
<tr>
<th>The Number of Selected Eigenfaces q</th>
<th>An Error Rate (%)</th>
<th>Processing Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q = 1$ (IPCA Algorithm)</td>
<td>24.44</td>
<td>The fastest</td>
</tr>
<tr>
<td>$q = 7$ (IPCA Algorithm)</td>
<td>18.33</td>
<td>Second</td>
</tr>
<tr>
<td>$q = 8$ (PCA Algorithm)</td>
<td>18.33</td>
<td>Third</td>
</tr>
<tr>
<td>$q = 2500$ (All Eigenvectors)</td>
<td>35.56</td>
<td>The slowest</td>
</tr>
</tbody>
</table>

5.2 Methods to Improve the Digital and the Optical Models

5.2.1 Introduction

In fact, the discussed models are not in their final stage where they can be optimized. Some ideas are presented in this section for each model that are going to help in enhancing their performance.

5.2.2 Improvement of the Digital and the Optical Models

The performance of face recognition and image detection of the digital model can be improved by increasing the size of the training faces and the detected or recognized unknown image. Also, increasing the size of the impulses and the detected or recognized unknown object of the optical model improves its performance in detection and recognition.

The performance of digital and optical recognition can be improved by obtaining a good way for blocking the pixels occupying the background of a face especially if the background is not black. If it is not black such as white, the error rate of recognition will increase because the intensities on the face will be close to 255 (i.e. they will be close to the intensities on the background) then discrimination between the intensities occupying the background and the intensities occupying the face becomes too hard.

For decreasing the error rates of face recognition and image detection of the digital model, the database of the training faces can be optimized in the same manner as the optimization of the database of the impulses for the optical model that is presented in Sub-sub-sub-subsection 2.3.4.3.2.1 and Sub-sub-sub-subsection 2.3.4.3.3.1.
The models performance can be improved by using another technique for enhancing the optimization speed of the databases. This technique measures information contained in the 180 tested images (or in the 108 objects) then picking the tested images (or the objects) that contain the highest information than the others to form the database of the training faces (or the database of the impulses).

Finally, the models performance can be improved by trying different detection and recognition thresholds such as thresholds generated by receiver operating characteristics (ROC).

### 5.3 The Digital Model Versus the Optical Model

In this section, we are going to compare between the digital model and the optical model in detection and recognition based on a couple of criteria. The comparison is summarized in Table 5.4.

<table>
<thead>
<tr>
<th>A Comparison Criterion</th>
<th>The Digital Model</th>
<th>The Optical Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Database</td>
<td>It is not necessarily to be optimized</td>
<td>It must be optimized</td>
</tr>
<tr>
<td>Implementation</td>
<td>Easier</td>
<td>Harder</td>
</tr>
<tr>
<td>Speed</td>
<td>Slower</td>
<td>Faster because it uses the speed of light</td>
</tr>
<tr>
<td>Detection and Recognition Performance</td>
<td>Better</td>
<td>Good</td>
</tr>
</tbody>
</table>

### 5.4 Future Work

For developing this work in future, the proposed ideas for improving the digital and the optical models presented in Section 5.2 are going to be achieved. Also, there is another idea that is considered to be performed in future is testing the models performance under various types of noises.
Bibliography


A Code for the Digital Model

This code is for testing face reconstruction, detection and recognition processes as well as the process of image compression by using principal components analysis (PCA) and improved principal components analysis (IPCA) algorithms. In addition to that this code is for setting up recognition and detection thresholds.

```matlab
% This Code Is for Testing Face Reconstruction, Detection and Recognition Processes as Well as the Process of Image Compression by Using Principal Components Analysis (PCA) and Improved Principal Components Analysis (IPCA) Algorithms. In Addition to That This Code Is for Setting up Recognition and Detection Thresholds.

clc
clear all
close all
format long

% Faces images are NxN images.
N=size(imread('Mr. Mansour Alshammari.jpg'),1); % This N is the number of pixels.

% Training faces.
Total_No_of_Known_Im=9; % The total number of the training faces.
All_Known_Im_V=zeros(N*N,Total_No_of_Known_Im); % An N^2xP, 2D matrix where P is the total number
```
Appendix A. A Code for the Digital Model

% of the training faces. Each training face is vectorized and placed in one of the columns of the 2D matrix. The size of each training face vector is \(N^2\).

Known_Images_Folder=.................
[cd '/The Known Images of Black Backgrounds']; % The folder of the black background training faces.

Known_Images_Folder=..........
[cd '/The Known Images of White Backgrounds']; % The folder of the white background training faces.

if isdir(Known_Images_Folder)==0
    Error_Message=sprintf('Error: The following folder does not exist\n%s',..............
    Known_Images_Folder);
    warndlg(Error_Message);
end

Known_Images=dir(fullfile(Known_Images_Folder,'*.jpg'));
for k=1:length(Known_Images)
    Known_Image=Known_Images(k).name;
    Known_Image_Location=fullfile(Known_Images_Folder,Known_Image);
    All_Known_Im_V(:,k)=................
    reshape(double(rgb2gray(imread(Known_Image_Location))).....
    ,N*N,1);

    % figure('units','centimeters','position',[16 7 7.5 8.5])
    % subplot(1,1,1)
    % imshow(uint8(reshape(All_Known_Im_V(:,k),N,N)))
    % if k==1
    %    title('Training Face No. num2str(k)');..............
    % elseif k==2
    %    title('Training Face No. num2str(k)');..............
    % elseif k==3
    %    title('Training Face No. num2str(k)');..............
    % elseif k==4
    %    title('Training Face No. num2str(k)');..............
else
    title('Training Face No. num2str(k)');..............
end

elseif k==3
    title({['Training Face No.' num2str(k)];.............
    ['( Known_Image(1:length(Known_Image)-6) ....
    ', 3^rd Projection')])
elseif k==4
    title({['Training Face No.' num2str(k)];.............
    ['( Known_Image(1:length(Known_Image)-6) ....
    ', 1^st Projection')])
elseif k==5
    title({['Training Face No.' num2str(k)];.............
    ['( Known_Image(1:length(Known_Image)-6) ....
    ', 2^nd Projection')])
elseif k==6
    title({['Training Face No.' num2str(k)];.............
    ['( Known_Image(1:length(Known_Image)-6) ....
    ', 3^rd Projection')])
elseif k==7
    title({['Training Face No.' num2str(k)];.............
    ['( Known_Image(1:length(Known_Image)-6) ....
    ', 1^st Projection')])
elseif k==8
    title({['Training Face No.' num2str(k)];.............
    ['( Known_Image(1:length(Known_Image)-6) ....
    ', 2^nd Projection')])
elseif k==9
    title({['Training Face No.' num2str(k)];.............
    ['( Known_Image(1:length(Known_Image)-6) ....
    ', 3^rd Projection')])
end

disp(['Please, press any keyboard button to explore ......
    'the remaining training faces >>>>>>>'])
pause
close all
clc

All_Known_Im_V;

Imhist for setting up a threshold to work on just the pixels of a
face and throwing the background pixels. Imhist calculates the
number of pixels in an image that have the same intensity levels.
So, if a training face has a unified background then the biggest
histogram of the intensity levels will be for the background pixels
because the total number of pixels that have the same
intensity levels are the background pixels of the training face.
Note that, the histogram of a digital image is defined as the
 discrete function, h(rk)=nk, where rk is the kth intensity level
and nk is the number of pixels in the image whose intensity level
is rk.
hist_Known_Im=zeros(Total_No_of_Known_Im,256);
for A=1:Total_No_of_Known_Im
% Note that, a training face has to be scaled between 0 to 255
% before using imhist. For doing that, uint8 can be used for
% converting the training face class form double to uint8.

hist_Known_Im(A,:) =.............
imhist(uint8(reshape(All_Known_Im_V(:,A),N,N)));

Known_Image=Known_Images(A).name;
plot(hist_Known_Im(A,:))
if A==1
    title({['The Histogram of Training Face No.' .......
        num2str(A)];........
    [' Known_Image(1:length(Known_Image)-6) ...........
      ', 1\textsuperscript{st} Projection']})
else A==2
    title({['The Histogram of Training Face No.' .......
        num2str(A)];......
    [' Known_Image(1:length(Known_Image)-6) ...........
      ', 2\textsuperscript{nd} Projection']})
else A==3
    title({['The Histogram of Training Face No.' .......
        num2str(A)];......
    [' Known_Image(1:length(Known_Image)-6) ...........
      ', 3\textsuperscript{rd} Projection']})
else A==4
    title({['The Histogram of Training Face No.' .......
        num2str(A)];......
    [' Known_Image(1:length(Known_Image)-6) ...........
      ', 1\textsuperscript{st} Projection']})
else A==5
    title({['The Histogram of Training Face No.' .......
        num2str(A)];......
    [' Known_Image(1:length(Known_Image)-6) ...........
      ', 2\textsuperscript{nd} Projection']})
else A==6
    title({['The Histogram of Training Face No.' .......
        num2str(A)];......
    [' Known_Image(1:length(Known_Image)-6) ...........
      ', 3\textsuperscript{rd} Projection']})
else A==7
    title({['The Histogram of Training Face No.' .......
        num2str(A)];......
    [' Known_Image(1:length(Known_Image)-6) ...........
      ', 1\textsuperscript{st} Projection']})
else A==8
    title({['The Histogram of Training Face No.' .......
        num2str(A)];......
    [' Known_Image(1:length(Known_Image)-6) ...........
      ', 2\textsuperscript{nd} Projection']})
else A==9
    title({['The Histogram of Training Face No.' .......

Appendix A. A Code for the Digital Model

177 \% num2str(A)];......
178 \% ['(' Known Image(1:length(Known Image)-6) ......
179 \% ', 3^{rd} Projection')]
180 \% end
181 \% xlabel('Intensity Level rk')
182 \% ylabel({'The Number of Pixels in the Training Face'.......
183 \% ' Whose Intensity Level Is rk Where h(rk)=nk'})
184 \% axis tight
185 \% disp(['Please, press any keyboard button to explore '......
186 \% 'the remaining histograms >>>>>>>'])
187 \% pause
188 \% clc
189 end
190 hist_Known_Im;
191
192 Mean_hist_Known_Im=sum(hist_Known_Im,1)/......
193 Total_No_of_Known_Im; \% The average histogram
194 \% for all training faces.
195 Threshold_Known_Im=8; \% The picked threshold is based on the
196 \% average histogram for all training faces
197 \% when the training faces have black
198 \% backgrounds. Note that, all intensity levels
199 \% below the threshold represent the images
200 \% backgrounds because these levels have the
201 \% biggest histogram.
202 \% Threshold_Known_Im=180; \% The picked threshold is based on the
203 \% average histogram for all training
204 \% faces when the training faces have
205 \% white backgrounds. Note that, all
206 \% intensity levels above the threshold
207 \% represent the images backgrounds
208 \% because these levels have the biggest
209 \% histogram.
210 \% plot(Mean_hist_Known_Im)
211 \% line([Threshold_Known_Im Threshold_Known_Im],......
212 \% [0 max(Mean_hist_Known_Im)],'Color','r')
213 \% text(Threshold_Known_Im+0.5,max(Mean_hist_Known_Im)/2,.....
214 \% '{color(red) The Threshold}')
215 \% title('The Mean Histogram of All Training Faces')
216 \% xlabel('Intensity Level rk')
217 \% set(gca,'XTick',[0 Threshold_Known_Im 255])
218 \% ylabel({'The Mean Number of Pixels from All Training' .........
219 \% ' Faces Whose Intensity Level Is rk'})
220 \% axis tight
221 \% pause
222
223 \% Normalizing all training faces for removing the lightening
224 \% effects on them and to increase the resolution of face detection
225 \% and recognition. Note that, the normalization will be done just
226 \% for face pixels for keeping the variations among the images just
% in the faces without the backgrounds effects.
Threshold_Known_Image=......
zeros(N,N,Total_No_of_Known_Im); % The training faces after
% applying the threshold.
Normalized_Known_Im_V=......
zeros(N*N,Total_No_of_Known_Im); % An N^2xP, 2D matrix where
% each column represents a
% normalized training face
% vector.
for M=1:Total_No_of_Known_Im
    t=reshape(All_Known_Im_V(:,M),N,N);
    T=t>Threshold_Known_Im; % The pixels bigger than the threshold
    % are of interest because they
    % represent the pixels of a face.
    % T=t<Threshold_Known_Im; % The pixels smaller than the
    % % threshold are of interest because
    % % they represent the pixels of
    % % a face.
    for R=1:N
        for C=1:N
            if T(R,C)==1
                Threshold_Known_Image(R,C,M)=......
                floor(255*(double(t(R,C))/......
                max(max(double(t))))); % The normalization of a
                % training face. This is
                % done to increase the
                % dynamic range of the
                % training face for
                % visualization by
                % scaling the
                % intensities of the
                % training face from 0
                % to 255.
            end;
        end;
    end;
    Normalized_Known_Im_V(:,M)=......
    reshape(Threshold_Known_Image(:, :, M), N*N, 1);
end;
% Known_Image=Known_Images(M).name;
figure
subplot(2,1,1)
imshow(t)
if M==1
    title({['This Is To Show How Good the Known '......
        'Images Threshold Is,'];......
    blanks(1);['Training Face No.' num2str(M)];
elseif M==2
    ['(' Known_Image(1:length(Known_Image)+6) ......
    ', 1^(st) Projection')]})
elseif M==2
% title({['This Is To Show How Good the Known '......
  'Images Threshold Is,'];......
blanks(1);['Training Face No.' num2str(M)];
['(' Known_Image(1:length(Known_Image)-6) .....'
', 2\{(nd Projection)']}])

else if M==3
  title({['This Is To Show How Good the Known '......
  'Images Threshold Is,'];......
blanks(1);['Training Face No.' num2str(M)];
['(' Known_Image(1:length(Known_Image)-6) .....'
', 3\{(rd Projection)']}])

else if M==4
  title({['This Is To Show How Good the Known '......
  'Images Threshold Is,'];......
blanks(1);['Training Face No.' num2str(M)];
['(' Known_Image(1:length(Known_Image)-6) .....'
', 1\{(st Projection)']}])

else if M==5
  title({['This Is To Show How Good the Known '......
  'Images Threshold Is,'];......
blanks(1);['Training Face No.' num2str(M)];
['(' Known_Image(1:length(Known_Image)-6) .....'
', 2\{(nd Projection)']}])

else if M==6
  title({['This Is To Show How Good the Known '......
  'Images Threshold Is,'];......
blanks(1);['Training Face No.' num2str(M)];
['(' Known_Image(1:length(Known_Image)-6) .....'
', 3\{(rd Projection)']}])

else if M==7
  title({['This Is To Show How Good the Known '......
  'Images Threshold Is,'];......
blanks(1);['Training Face No.' num2str(M)];
['(' Known_Image(1:length(Known_Image)-6) .....'
', 1\{(st Projection)']}])

else if M==8
  title({['This Is To Show How Good the Known '......
  'Images Threshold Is,'];......
blanks(1);['Training Face No.' num2str(M)];
['(' Known_Image(1:length(Known_Image)-6) .....'
', 2\{(nd Projection)']}])

else if M==9
  title({['This Is To Show How Good the Known '......
  'Images Threshold Is,'];......
blanks(1);['Training Face No.' num2str(M)];
['(' Known_Image(1:length(Known_Image)-6) .....'
', 3\{(rd Projection)']}])

end

subplot(2,1,2)
imshow(reshape(Normalized_Known_Im_V(:,M),N,N))
if M==1
    title(['Normalized Training Face No.' num2str(M)];
    ['(' Known_Image(1:length(Known_Image)-6) ......
    ', 1^{st} Projection')'])
elseif M==2
    title(['Normalized Training Face No.' num2str(M)];
    ['(' Known_Image(1:length(Known_Image)-6) ......
    ', 2^{nd} Projection')'])
elseif M==3
    title(['Normalized Training Face No.' num2str(M)];
    ['(' Known_Image(1:length(Known_Image)-6) ......
    ', 3^{rd} Projection')'])
elseif M==4
    title(['Normalized Training Face No.' num2str(M)];
    ['(' Known_Image(1:length(Known_Image)-6) ......
    ', 1^{st} Projection')'])
elseif M==5
    title(['Normalized Training Face No.' num2str(M)];
    ['(' Known_Image(1:length(Known_Image)-6) ......
    ', 2^{nd} Projection')'])
elseif M==6
    title(['Normalized Training Face No.' num2str(M)];
    ['(' Known_Image(1:length(Known_Image)-6) ......
    ', 3^{rd} Projection')'])
elseif M==7
    title(['Normalized Training Face No.' num2str(M)];
    ['(' Known_Image(1:length(Known_Image)-6) ......
    ', 1^{st} Projection')'])
elseif M==8
    title(['Normalized Training Face No.' num2str(M)];
    ['(' Known/Image(1:length(Known/Image)-6) ......
    ', 2^{nd} Projection')'])
elseif M==9
    title(['Normalized Training Face No.' num2str(M)];
    ['(' Known/Image(1:length(Known/Image)-6) ......
    ', 3^{rd} Projection')'])
end
disp(['Please, press any keyboard button to see how ' ......
'good the applied'])
disp('threshold on the normalized training faces is >>>>>>>')
pause
close all
cic
figure
subplot(2,1,1)
imshow(uint8(t))
if M==1
    title(['Training Face No.' num2str(M)];
    ['(' Known/Image(1:length(Known/Image)-6) ......
Appendix A. A Code for the Digital Model

```matlab
% , 1^{st} Projection')})
elseif M==2
    title({['Training Face No.' num2str(M)];........
     ['( Known_Image(1:length(Known_Image)-6) ....
     ', 2^{nd} Projection')])
elseif M==3
    title({['Training Face No.' num2str(M)];........
     ['( Known_Image(1:length(Known_Image)-6) ....
     ', 3^{rd} Projection')])
elseif M==4
    title({['Training Face No.' num2str(M)];........
     ['( Known_Image(1:length(Known_Image)-6) ....
     ', 1^{st} Projection')])
elseif M==5
    title({['Training Face No.' num2str(M)];........
     ['( Known_Image(1:length(Known_Image)-6) ....
     ', 2^{nd} Projection')])
elseif M==6
    title({['Training Face No.' num2str(M)];........
     ['( Known_Image(1:length(Known_Image)-6) ....
     ', 3^{rd} Projection')])
elseif M==7
    title({['Training Face No.' num2str(M)];........
     ['( Known_Image(1:length(Known_Image)-6) ....
     ', 1^{st} Projection')])
elseif M==8
    title({['Training Face No.' num2str(M)];........
     ['( Known_Image(1:length(Known_Image)-6) ....
     ', 2^{nd} Projection')])
elseif M==9
    title({['Training Face No.' num2str(M)];........
     ['( Known_Image(1:length(Known_Image)-6) ....
     ', 3^{rd} Projection')])
end
subplot(2,1,2)
imshow(uint8(reshape(Normalized_Known_Im_V(:,M),N,N)))
if M==1
    title({['Normalized Training Face No.' num2str(M)];....
     ['( Known_Image(1:length(Known_Image)-6) ....
     ', 1^{st} Projection')])
elseif M==2
    title({['Normalized Training Face No.' num2str(M)];....
     ['( Known_Image(1:length(Known_Image)-6) ....
     ', 2^{nd} Projection')])
elseif M==3
    title({['Normalized Training Face No.' num2str(M)];....
     ['( Known_Image(1:length(Known_Image)-6) ....
     ', 3^{rd} Projection')])
elseif M==4
    title({['Normalized Training Face No.' num2str(M)];....
```
% 
% elseif M==5
  title({"Normalized Training Face No." num2str(M);... 
    " Known_Image(1:length(Known_Image)-6) ...... 
    ', 1^{st} Projection')})
% elseif M==6
  title({"Normalized Training Face No." num2str(M);... 
    " Known_Image(1:length(Known_Image)-6) ...... 
    ', 2^{nd} Projection')})
% elseif M==7
  title({"Normalized Training Face No." num2str(M);... 
    " Known_Image(1:length(Known_Image)-6) ...... 
    ', 3^{rd} Projection')})
% elseif M==8
  title({"Normalized Training Face No." num2str(M);... 
    " Known_Image(1:length(Known_Image)-6) ...... 
    ', 2^{nd} Projection')})
% elseif M==9
  title({"Normalized Training Face No." num2str(M);... 
    " Known_Image(1:length(Known_Image)-6) ...... 
    ', 3^{rd} Projection')})
end
% disp(['Please, press any keyboard button to explore ...... 
    'the remaining normalized training faces >>>>>>>'])
% pause
% close all
% clc
Normalized_Known_Im_V;

% Centering each face by simply subtracting the mean of the face 
% pixels from each pixel in the face. By doing that the new face 
% pixels will have zero mean that means the face is centered.
Rows_Columns=zeros(1,1,.............
  Total_No_of_Known_Im); % The rows and columns for the pixels 
  % of the faces. Note that, the training 
  % faces are not similar so the rows and 
  % columns of the faces pixels will not 
  % be equal. Therefore, MATLAB will add 
  % zero rows and columns to make the 
  % matrices of the rows and columns of 
  % the faces pixels are equal.
Means=zeros(1,Total_No_of_Known_Im); % The means of the 
  % faces pixels.
Centered_Known_Im=zeros(N*N,.............
  Total_No_of_Known_Im); % An N^2xF, 2D matrix where each 
  % column represents a training 
  % face vector with a centered 
  % face.
Appendix A. A Code for the Digital Model

```matlab
477   Centered_Known_Image=zeros(N,N,........
478       Total_No_of_Known_Im); % The training faces after
479       % centering the faces.
480   for j=1:Total_No_of_Known_Im
481       x=reshape(Normalized_Known_Im_V(:,j),N,N);
482       [rr cc]=find(x>0); % The pixels bigger than zero are of
483       % interest because they represent the
484       % pixels of a face.
485       Rows_Columns(1,1:size(rr,1),j)=rr.';
486       Rows_Columns(2,1:size(rr,1),j)=cc.';
487       Sum=0;
488       No=0;
489       for RR=1:size(rr,1)
490           Sum=Sum+x(rr(RR),cc(RR));
491           No=No+1;
492       end
493       Means(1,j)=Sum/No;
494
495       for RR1=1:size(rr,1)
496           Centered_Known_Image(rr(RR1),cc(RR1),j)=........
497                        x(rr(RR),cc(RR))−Means(1,j);
498       end
499   Centered_Known_Im(:,j)=........
500       reshape(Centered_Known_Image(:,:,j),N*N,1);
501
502   % figure('units','centimeters','position',[16 7 7 8.5])
503   % subplot(1,1,1)
504   % Known_Image=Known_Images(j).name;
505   % imshow(uint8(reshape(Centered_Known_Im(:,j),N,N)))
506   % if j==1
507       title({['Centered Training Face No.' num2str(j)];....
508           [' ', 1\^st Projection]})
509   % elseif j==2
510       title({['Centered Training Face No.' num2str(j)];....
511           [' ', 2\^nd Projection]})
512   % elseif j==3
513       title({['Centered Training Face No.' num2str(j)];....
514           [' ', 3\^rd Projection]})
515   % elseif j==4
516       title({['Centered Training Face No.' num2str(j)];....
517           [' ', 1\^st Projection]})
518   % elseif j==5
519       title({['Centered Training Face No.' num2str(j)];....
520           [' ', 2\^{nd} Projection]})
521   % elseif j==6
```
% title({['Centered Training Face No.' num2str(j)];....
% [', 3^{rd} Projection']})
elseif j==7
    title({['Centered Training Face No.' num2str(j)];....
% [', 1^{st} Projection']})
elseif j==8
    title({['Centered Training Face No.' num2str(j)];....
% [', 2^{nd} Projection']})
end

% disp(['Please, press any keyboard button to explore '......
% 'the remaining centered training faces >>>>>>>'])
pause
close all
cic

% Tested images are supposed to be unknown but here multiple
% images for each training face are taken for testing the face
% reconstruction, recognition and detection processes as well
% as selecting a decision threshold for face detection and
% recognition.
Total_No_of_Tested_Im=180; % Total number of the
% tested images.

Im_P=[36 12 12 36 12 12 36 12 12]; % Each element in this vector
  % represents the total number of
  % the taken images for each
  % training face.

L1=Im_P(1);
L2=L1+Im_P(2);
L3=L2+Im_P(3); % L3=60 is the total number of the tested
  % images for Mr. Mansour Alshammari.
L4=L3+Im_P(4);
L5=L4+Im_P(5);
L6=L5+Im_P(6); % L6=120 is the total number of the tested images
  % for Mr. Methkir Alharthee.
L7=L6+Im_P(7);
L8=L7+Im_P(8);
L9=L8+Im_P(9); % L9=180 is the total number of the tested images
  % for Mr. Mohammed Hanafy.
All_Tested_Im_V=zeros(N*N,............
Total_No_of_Tested_Im); % An N^2xP1, 2D matrix where P1 is the
% total number of the tested images.
% Each tested image is vectorized and
% placed in one of the columns of the
% 2D matrix. The size of each tested
% image vector is N^2.
Tested_Images_Folder=............
[cd '/The Tested Images of Black Backgrounds']; % The folder of
% the black
% background
% tested
% images.
% Tested_Images_Folder=............
% [cd '/The Tested Images of White Backgrounds']; % The folder
% of the white
% background
% tested
% images.

if isdir(Tested_Images_Folder)==0
    Error_Message1=sprintf(.........
    'Error: The following folder does not exist\n%s'.......         
    ,Tested_Images_Folder);
    warndlg(Error_Message1);
end

Tested_Images=dir(fullfile(Tested_Images_Folder,'*.jpg'));
for k1=1:length(Tested_Images)
    Tested_Image_Number=[num2str(k1) '.jpg'];
    Tested_Image_Location=............
    fullfile(Tested_Images_Folder,Tested_Image_Number);
    All_Tested_Im_V(:,k1)=reshape(............
    (double(rgb2gray(imread(Tested_Image_Location))),N*N,1);
    % figure('units','centimeters','position',\[16 7 7.5 8.5])
    % subplot(1,1,1)
    % imshow(uint8(reshape(All_Tested_Im_V(:,k1),N,N)))
    % if k1<=L1
      % title({'Tested Image No.' num2str(k1)});
      % '(Mr. Mansour Alshammari, 1^{st} Projection}')
    % elseif k1>L1 && k1<=L2
      % title({'Tested Image No.' num2str(k1)});
      % '(Mr. Mansour Alshammari, 2^{nd} Projection}')
    % elseif k1>L2 && k1<=L3
      % title({'Tested Image No.' num2str(k1)});
      % '(Mr. Mansour Alshammari, 3^{rd} Projection}')
    % elseif k1>L3 && k1<=L4
title({['Tested Image No.' num2str(k1)];...........
'Mr. Methkir Alharthee, 1\(^{st}\) Projection')})
elseif k1>L4 && k1<=L5
    title({['Tested Image No.' num2str(k1)];...........
'Mr. Methkir Alharthee, 2\(^{nd}\) Projection')})
elseif k1>L5 && k1<=L6
    title({['Tested Image No.' num2str(k1)];...........
'Mr. Methkir Alharthee, 3\(^{rd}\) Projection')})
elseif k1>L6 && k1<=L7
    title({['Tested Image No.' num2str(k1)];...........
'Mr. Mohammed Hanafy, 1\(^{st}\) Projection')})
elseif k1>L7 && k1<=L8
    title({['Tested Image No.' num2str(k1)];...........
'Mr. Mohammed Hanafy, 2\(^{nd}\) Projection')})
elseif k1>L8 && k1<=L9
    title({['Tested Image No.' num2str(k1)];...........
'Mr. Mohammed Hanafy, 3\(^{rd}\) Projection')})
end
disp(['Please, press any keyboard button to explore '......
'the remaining tested images >>>>>>>'])
pause
close all
cic

All_Tested_Im_V;

% Imhist for setting up a threshold to work on just the pixels of a
% face and throwing the background pixels. Imhist calculates the
% number of pixels in an image that have the same intensity levels.
% So, if a tested image has a unified background then the biggest
% histogram of the intensity levels will be for the background
% pixels because the total number of pixels that have the same
% intensity levels are the background pixels of the tested image.
% Note that, the histogram of a digital image is defined as the
% discrete function, h(rk)=nk, where rk is the kth intensity level
% and nk is the number of pixels in the image whose intensity level
% is rk.

hist_Tested_Im=zeros(Total_No_of_Tested_Im,256);
for A1=1:Total_No_of_Tested_Im
    % Note that, a tested image has to be scaled between 0 to 255
    % before using imhist. For doing that, uint8 can be used for
    % converting the tested image class form double to uint8.
    hist_Tested_Im(A1,:)=........
    imhist(uint8(reshape(All_Tested_Im_V(:,A1),N,N)));   
    plot(hist_Tested_Im(A1,:))
    if A1<=L1
        title({['The Histogram of Training Face No.' ......
            num2str(A1)];........
            'Mr. Mansour Alshammari, 1\(^{st}\) Projection')}})
elseif A1>L1 && A1<=L2
    title({['The Histogram of Training Face No.' .......
          num2str(A1)];........
          '(Mr. Mansour Alshammari, 2^{nd} Projection')})
elseif A1>L2 && A1<=L3
    title({['The Histogram of Training Face No.' .......
          num2str(A1)];........
          '(Mr. Mansour Alshammari, 3^{rd} Projection')})
elseif A1>L3 && A1<=L4
    title({['The Histogram of Training Face No.' .......
          num2str(A1)];........
          '(Mr. Methkir Alharthee, 1^{st} Projection')})
elseif A1>L4 && A1<=L5
    title({['The Histogram of Training Face No.' .......
          num2str(A1)];........
          '(Mr. Methkir Alharthee, 2^{nd} Projection')})
elseif A1>L5 && A1<=L6
    title({['The Histogram of Training Face No.' .......
          num2str(A1)];........
          '(Mr. Methkir Alharthee, 3^{rd} Projection')})
elseif A1>L6 && A1<=L7
    title({['The Histogram of Training Face No.' .......
          num2str(A1)];........
          '(Mr. Mohammed Hanafy, 1^{st} Projection')})
elseif A1>L7 && A1<=L8
    title({['The Histogram of Training Face No.' .......
          num2str(A1)];........
          '(Mr. Mohammed Hanafy, 2^{nd} Projection')})
elseif A1>L8 && A1<=L9
    title({['The Histogram of Training Face No.' .......
          num2str(A1)];........
          '(Mr. Mohammed Hanafy, 3^{rd} Projection')})
end
xlabel('Intensity Level rk')
ylabel({'The Number of Pixels in the Tested Image' ........
       ' Whose Intensity Level Is rk Where h(rk)=nk'})
axis tight
disp(['Please, press any keyboard button to explore ' ......
     'the remaining histograms >>>>>>>'])
pause
clc

hist_Tested_Im;

Mean_hist_Tested_Im=............
sum(hist_Tested_Im,1)/Total_No_of_Tested_Im; % The average
% histogram for
% all tested
% images.
Threshold_Tested_Im=8; % The picked threshold is based on the
% average histogram for all tested images
% when the tested images have black
% backgrounds. Note that, all intesity
% levels below the threshold represent the
% images backgrounds because these levels
% have the biggest histogram.

% Threshold_Tested_Im=180; % The picked threshold is based on the
% average histogram for all tested
% images when the tested images have
% white backgrounds. Note that, all
% intensity levels above the threshold
% represent the images backgrounds
% because these levels have the biggest
% histogram.

% plot(Mean_hist_Tested_Im)
% line([Threshold_Tested_Im Threshold_Tested_Im]............
% [0 max(Mean_hist_Tested_Im)],'Color','r')
% text(Threshold_Tested_Im+0.5,max(Mean_hist_Tested_Im)/2,........
% '{\color{red} The Threshold}')
% title('The Mean Histogram of All Tested Images')
% xlabel('Intensity Level rk')
% set(gca,'XTick',[0 Threshold_Tested_Im 255])
% ylabel({'The Mean Number of Pixels from All Tested'......
% 'Images Whose Intensity Level Is rk'})
% axis tight
% pause

% Normalizing all the tested images for removing the lightening
% effects on them and to increase the resolution of face detection
% and recognition. Note that, the normalization will be done just
% for face pixels for keeping the variations among the images just
% in the faces without the backgrounds effects.

Threshold_Tested_Im=...........
zeros(N,N,Total_No_of_Tested_Im); % The tested images after
% applying the threshold.

Normalized_Tested_Im_V=zeros(N*N,...........
Total_No_of_Tested_Im); % An N^2xP1, 2D matrix where each
% column represents a normalized
% tested image vector.

for M1=1:Total_No_of_Tested_Im
    t1=reshape(All_Tested_Im_V(:,M1),N,N);
    T1=t1>Threshold_Tested_Im; % The pixels bigger than the
% threshold are of interest because
% they represent the pixels of
% a face.
    T1=t1<Threshold_Tested_Im; % The pixels smaller than the
% threshold are of interest
% because they represent the
% pixels of a face.
for C1=1:N
    if T1(R1,C1)==1
        Threshold_Tested_Image(R1,C1,M1)=.............
        floor(255*(double(t1(R1,C1))/........
        max(max(double(t1))))); % The normalization of
        % a tested image. This
        % is done to increase
        % the dynamic range of
        % the tested image for
        % visualization by
        % scaling the
        % intensities of the
        % tested image from 0
        % to 255.
    end;
end;
end;

Normalized_Tested_Im_V(:,M1)=..............
reshape(Threshold_Tested_Image(:,:,M1),N*N,1);

% figure
% subplot(2,1,1)
% imshow(t1)
if M1<=L1
    title({['This Is To Show How Good the Tested '.....
    'Images Threshold Is,'];.....
    blanks(1);['Tested Image No.' num2str(M1)];.....
    '(Mr. Mansour Alshammari, 1^{st} Projection')})
elseif M1>L1 && M1<=L2
    title({['This Is To Show How Good the Tested '.....
    'Images Threshold Is,'];.....
    blanks(1);['Tested Image No.' num2str(M1)];.....
    '(Mr. Mansour Alshammari, 2^{nd} Projection')})
elseif M1>L2 && M1<=L3
    title({['This Is To Show How Good the Tested '.....
    'Images Threshold Is,'];.....
    blanks(1);['Tested Image No.' num2str(M1)];.....
    '(Mr. Mansour Alshammari, 3^{rd} Projection')})
elseif M1>L3 && M1<=L4
    title({['This Is To Show How Good the Tested '.....
    'Images Threshold Is,'];.....
    blanks(1);['Tested Image No.' num2str(M1)];.....
    '(Mr. Methkir Alharthee, 1^{st} Projection')})
elseif M1>L4 && M1<=L5
    title({['This Is To Show How Good the Tested '.....
    'Images Threshold Is,'];.....
    blanks(1);['Tested Image No.' num2str(M1)];.....
    '(Mr. Methkir Alharthee, 2^{nd} Projection')})
elseif M1>L5 && M1<=L6
    title({['This Is To Show How Good the Tested '.....
% 'Images Threshold Is,'];......
% blanks(1);['Tested Image No.' num2str(M1)];......
% '(Mr. Methkir Alharthee, 3\textsuperscript{rd} Projection')}
elsif M1>L6 && M1<=L7
  title({'This Is To Show How Good the Tested '....
% 'Images Threshold Is,'];......
% blanks(1);['Tested Image No.' num2str(M1)];......
% '(Mr. Mohammed Hanafy, 1\textsuperscript{st} Projection')}
elsif M1>L7 && M1<=L8
  title({'This Is To Show How Good the Tested '....
% 'Images Threshold Is,'];......
% blanks(1);['Tested Image No.' num2str(M1)];......
% '(Mr. Mohammed Hanafy, 2\textsuperscript{nd} Projection')}
elsif M1>L8 && M1<=L9
  title({'This Is To Show How Good the Tested '....
% 'Images Threshold Is,'];......
% blanks(1);['Tested Image No.' num2str(M1)];......
% '(Mr. Mohammed Hanafy, 3\textsuperscript{rd} Projection')}
end
subplot(2,1,2)
imshow(reshape(Normalized_Tested_Im_V(:,M1),N,N))
if M1<=L1
  title({'Normalized Tested Image No.' num2str(M1)];....
% '(Mr. Mansour Alshammari, 1\textsuperscript{st} Projection')}
elsif M1>L1 && M1<=L2
  title({'Normalized Tested Image No.' num2str(M1)];....
% '(Mr. Mansour Alshammari, 2\textsuperscript{nd} Projection')}
elsif M1>L2 && M1<=L3
  title({'Normalized Tested Image No.' num2str(M1)];....
% '(Mr. Mansour Alshammari, 3\textsuperscript{rd} Projection')}
elsif M1>L3 && M1<=L4
  title({'Normalized Tested Image No.' num2str(M1)];....
% '(Mr. Methkir Alharthee, 1\textsuperscript{st} Projection')}
elsif M1>L4 && M1<=L5
  title({'Normalized Tested Image No.' num2str(M1)];....
% '(Mr. Methkir Alharthee, 2\textsuperscript{nd} Projection')}
elsif M1>L5 && M1<=L6
  title({'Normalized Tested Image No.' num2str(M1)];....
% '(Mr. Methkir Alharthee, 3\textsuperscript{rd} Projection')}
elsif M1>L6 && M1<=L7
  title({'Normalized Tested Image No.' num2str(M1)];....
% '(Mr. Mohammed Hanafy, 1\textsuperscript{st} Projection')}
elsif M1>L7 && M1<=L8
  title({'Normalized Tested Image No.' num2str(M1)];....
% '(Mr. Mohammed Hanafy, 2\textsuperscript{nd} Projection')}
elsif M1>L8 && M1<=L9
  title({'Normalized Tested Image No.' num2str(M1)];....
% '(Mr. Mohammed Hanafy, 3\textsuperscript{rd} Projection')}
end
disp({'Please, press any keyboard button to '......
Appendix A. A Code for the Digital Model

877 % 'see how good the applied']
878 % disp('threshold on the normalized tested images is >>>>>>>')
879 % pause
880 % close all
881 % clc
882 % figure
883 % subplot(2,1,1)
884 % imshow(uint8(t1))
885 % if M1<=L1
886 % title({['Tested Image No.' num2str(M1)];........
887 % '(Mr. Mansour Alshammari, 1^{st} Projection')})
888 % elseif M1>L1 && M1<=L2
889 % title({['Tested Image No.' num2str(M1)];........
890 % '(Mr. Mansour Alshammari, 2^{nd} Projection')})
891 % elseif M1>L2 && M1<=L3
892 % title({['Tested Image No.' num2str(M1)];........
893 % '(Mr. Mansour Alshammari, 3^{rd} Projection')})
894 % elseif M1>L3 && M1<=L4
895 % title({['Tested Image No.' num2str(M1)];........
896 % '(Mr. Methkir Alharthee, 1^{st} Projection')})
897 % elseif M1>L4 && M1<=L5
898 % title({['Tested Image No.' num2str(M1)];........
899 % '(Mr. Methkir Alharthee, 2^{nd} Projection')})
900 % elseif M1>L5 && M1<=L6
901 % title({['Tested Image No.' num2str(M1)];........
902 % '(Mr. Methkir Alharthee, 3^{rd} Projection')})
903 % elseif M1>L6 && M1<=L7
904 % title({['Tested Image No.' num2str(M1)];........
905 % '(Mr. Mohammed Hanafy, 1^{st} Projection')})
906 % elseif M1>L7 && M1<=L8
907 % title({['Tested Image No.' num2str(M1)];........
908 % '(Mr. Mohammed Hanafy, 2^{nd} Projection')})
909 % elseif M1>L8 && M1<=L9
910 % title({['Tested Image No.' num2str(M1)];........
911 % '(Mr. Mohammed Hanafy, 3^{rd} Projection')})
912 % end
913 % subplot(2,1,2)
914 % imshow(uint8(reshape(Normalized_Tested_Im_V(:,M1),N,N)))
915 % if M1<=L1
916 % title({['Normalized Tested Image No.' num2str(M1)];....
917 % '(Mr. Mansour Alshammari, 1^{st} Projection')})
918 % elseif M1>L1 && M1<=L2
919 % title({['Normalized Tested Image No.' num2str(M1)];....
920 % '(Mr. Mansour Alshammari, 2^{nd} Projection')})
921 % elseif M1>L2 && M1<=L3
922 % title({['Normalized Tested Image No.' num2str(M1)];....
923 % '(Mr. Mansour Alshammari, 3^{rd} Projection')})
924 % elseif M1>L3 && M1<=L4
925 % title({['Normalized Tested Image No.' num2str(M1)];....
926 % (Mr. Mansour Alshammari, 1^{st} Projection')})
% (Mr. Methkir Alharthee, 1\textsuperscript{st} Projection})
elseif M1>L4 && M1<=L5
    title({['Normalized Tested Image No.' num2str(M1)];....
% (Mr. Methkir Alharthee, 2\textsuperscript{nd} Projection})
elseif M1>L5 && M1<=L6
    title({['Normalized Tested Image No.' num2str(M1)];....
% (Mr. Methkir Alharthee, 3\textsuperscript{rd} Projection})
elseif M1>L6 && M1<=L7
    title({['Normalized Tested Image No.' num2str(M1)];....
% (Mr. Mohammed Hanafy, 1\textsuperscript{st} Projection})
elseif M1>L7 && M1<=L8
    title({['Normalized Tested Image No.' num2str(M1)];....
% (Mr. Mohammed Hanafy, 2\textsuperscript{nd} Projection})
elseif M1>L8 && M1<=L9
    title({['Normalized Tested Image No.' num2str(M1)];....
% (Mr. Mohammed Hanafy, 3\textsuperscript{rd} Projection})
end
disp(['Please, press any keyboard button to explore '....
' the remaining normalized tested images >>>>>>>'])
pause
close all
cic
Normalized_Tested_Im_V;

% Centering each face by simply subtracting the mean of the face
% pixels from each pixel in the face. By doing that the new face
% pixels will have zero mean that means the face is centered.
Total_No_of_Tested_Im); % The rows and columns for the
% pixels of the faces. Note that,
% the tested images are not similar
% so the rows and columns of the
% faces pixels will not be equal.
% Therefore, MATLAB will add zero
% rows and columns to make the
% matrices of the rows and columns
% of the faces pixels are equal.
Means1=zeros(1,Total_No_of_Tested_Im);
Centered_Tested_Im=..........
zeros(N*N,Total_No_of_Tested_Im); % An N^2xP1, 2D matrix where
% each column represents a
% tested image vector with a
% centered face.
Centered_Tested_Image=..........
zeros(N,N,Total_No_of_Tested_Im); % The tested images after
% centering the faces.
for j1=1:Total_No_of_Tested_Im
    x1=reshape(Normalized_Tested_Im_V(:,j1),N,N);
    [rr1 cc1]=find(x1>0); % The pixels bigger than zero are of
% interest because they represent the
% pixels of the faces.

Rows_Columns1(1,1:size(rr1,1),j1)=rr1.'.;
Rows_Columns1(2,1:size(rr1,1),j1)=cc1.'.;
Sum1=0;
No1=0;
for RR2=1:size(rr1,1)
    Sum1=Sum1+x1(rr1(RR2),cc1(RR2));
    No1=No1+1;
end

Means1(1,j1)=Sum1/No1;
for RR3=1:size(rr1,1)
    Centered_Tested_Image(rr1(RR3),cc1(RR3),j1)=........
    x1(rr1(RR3),cc1(RR3))−Means1(1,j1);
end
Centered_Tested_Im(:,j1)=........
reshape(Centered_Tested_Image(:,:,j1),N*N,1);

% figure('units','centimeters','position',[16 7 7 8.5])
% subplot(1,1,1)
% imshow(uint8(reshape(Centered_Tested_Im(:,j1),N,N)))
% if j1<=L1
%    title({'Centered Tested Image No.' num2str(j1);.....
%    '(Mr. Mansour Alshammari, 1^{st} Projection')}))
% elseif j1>L1 && j1<=L2
%    title({'Centered Tested Image No.' num2str(j1);.....
%    '(Mr. Mansour Alshammari, 2^{nd} Projection')}))
% elseif j1>L2 && j1<=L3
%    title({'Centered Tested Image No.' num2str(j1);.....
%    '(Mr. Mansour Alshammari, 3^{rd} Projection')}))
% elseif j1>L3 && j1<=L4
%    title({'Centered Tested Image No.' num2str(j1);.....
%    '(Mr. Methkir Alharthee, 1^{st} Projection')}))
% elseif j1>L4 && j1<=L5
%    title({'Centered Tested Image No.' num2str(j1);.....
%    '(Mr. Methkir Alharthee, 2^{nd} Projection')}))
% elseif j1>L5 && j1<=L6
%    title({'Centered Tested Image No.' num2str(j1);.....
%    '(Mr. Methkir Alharthee, 3^{rd} Projection')}))
% elseif j1>L6 && j1<=L7
%    title({'Centered Tested Image No.' num2str(j1);.....
%    '(Mr. Mohammed Hanafy, 1^{st} Projection')}))
% elseif j1>L7 && j1<=L8
%    title({'Centered Tested Image No.' num2str(j1);.....
%    '(Mr. Mohammed Hanafy, 2^{nd} Projection')}))
% elseif j1>L8 && j1<=L9
%    title({'Centered Tested Image No.' num2str(j1);.....
%    '(Mr. Mohammed Hanafy, 3^{rd} Projection')}))
% end
% disp(['Please, press any keyboard button to explore ' ......
' the remaining centered tested images >>>>>>>'])

% pause
% close all
end

Centered_Tested_Im;

% Centering the set of the training faces by simply subtracting the
% mean training face from each training face in the set. By doing
% that the new set of the training faces will have zero mean which
% means the set is centered.
Av_Image=sum(Centered_Known_Im,2)./......
Total_No_of_Known_Im; % The average training face.

% figure('units','centimeters','position',[12 4 12.5 13])
% subplot(1,1,1)
% Reshaped_Av_Image=reshape(Av_Image,N,N);
% Negative_Av_Image=255*ones(N,N)-255*(Reshaped_Av_Image/......
% max(max(Reshaped_Av_Image))); % Obtaining a negative image
% % for the mean training face
% % in order to enhance its
% % appearance.
% imshow(uint8(Negative_Av_Image))
% title('The Average Training Face')
% pause
% close all

Known_Im_Subt_Mean=......
zeros(N*N,Total_No_of_Known_Im); % An N^2xP, 2D matrix where
% each column represents a
% centered (with respect to
% the set of the training
% faces) training face vector.

for J=1:Total_No_of_Known_Im

    Known_Im_Subt_Mean(:,J)=Centered_Known_Im(:,J)-Av_Image;
    Reshaped_Known_Im_Subt_Mean=......
    reshape(Known_Im_Subt_Mean(:,J),N,N);
    Negative_Known_Im_Subt_Mean=255*ones(N,N)-......
    255*(Reshaped_Known_Im_Subt_Mean/max(......
    Reshaped_Known_Im_Subt_Mean)); % Obtaining a negative
% image for the centered
% training face in order
% to enhance its
% appearance.

Known_Image=Known_Images(J).name;
% figure('units','centimeters','position',[12 4 12.5 14])
% subplot(1,1,1)
% imshow(uint8(Negative_Known_Im_Subt_Mean))
% if J==1
    title({['Centered (with Respect to the Set of the ' ....
    'Training Faces) Training Face No.' num2str(J)]; ....
    ['(' Known_Image(1:length(Known_Image)-6) .......
    ', 1\textsuperscript{st} Projection)'; '(A Negative Image)']})
else J==2
    title({['Centered (with Respect to the Set of the ' ....
    'Training Faces) Training Face No.' num2str(J)]; ....
    ['(' Known_Image(1:length(Known_Image)-6) .......
    ', 2\textsuperscript{nd} Projection)'; '(A Negative Image)']})
else J==3
    title({['Centered (with Respect to the Set of the ' ....
    'Training Faces) Training Face No.' num2str(J)]; ....
    ['(' Known_Image(1:length(Known_Image)-6) .......
    ', 3\textsuperscript{rd} Projection)'; '(A Negative Image)']})
else J==4
    title({['Centered (with Respect to the Set of the ' ....
    'Training Faces) Training Face No.' num2str(J)]; ....
    ['(' Known_Image(1:length(Known_Image)-6) .......
    ', 1\textsuperscript{st} Projection)'; '(A Negative Image)']})
else J==5
    title({['Centered (with Respect to the Set of the ' ....
    'Training Faces) Training Face No.' num2str(J)]; ....
    ['(' Known_Image(1:length(Known_Image)-6) .......
    ', 2\textsuperscript{nd} Projection)'; '(A Negative Image)']})
else J==6
    title({['Centered (with Respect to the Set of the ' ....
    'Training Faces) Training Face No.' num2str(J)]; ....
    ['(' Known_Image(1:length(Known_Image)-6) .......
    ', 3\textsuperscript{rd} Projection)'; '(A Negative Image)']})
else J==7
    title({['Centered (with Respect to the Set of the ' ....
    'Training Faces) Training Face No.' num2str(J)]; ....
    ['(' Known_Image(1:length(Known_Image)-6) .......
    ', 1\textsuperscript{st} Projection)'; '(A Negative Image)']})
else J==8
    title({['Centered (with Respect to the Set of the ' ....
    'Training Faces) Training Face No.' num2str(J)]; ....
    ['(' Known_Image(1:length(Known_Image)-6) .......
    ', 2\textsuperscript{nd} Projection)'; '(A Negative Image)']})
else J==9
    title({['Centered (with Respect to the Set of the ' ....
    'Training Faces) Training Face No.' num2str(J)]; ....
    ['(' Known_Image(1:length(Known_Image)-6) .......
    ', 3\textsuperscript{rd} Projection)'; '(A Negative Image)']})
end
disp(['Please, press any keyboard button to explore ' ....
      'the remaining centered training faces >>>>>>>'])
pause
% close all
% clc

dend

Known_Im_Subt_Mean;

Tested_Im_Subt_Mean=.........
zeros(N*N,Total_No_of_Tested_Im); % An N^2xP1, 2D matrix where
% each column represents a
% tested image vector after
% subtracting the average
% training face.

for J1=1:Total_No_of_Tested_Im

Tested_Im_Subt_Mean(:,J1)=Centered_Tested_Im(:,J1)-Av_Image;
Reshaped_Tested_Im_Subt_Mean=.........
reshape(Tested_Im_Subt_Mean(:,J1),N,N);

Negative_Tested_Im_Subt_Mean=255*ones(N,N)-255*........
(Reshaped_Tested_Im_Subt_Mean/max(max(........
Reshaped_Tested_Im_Subt_Mean))); % Obtaining a negative
% image for the tested
% image in order to
% enhance its appearance.

figure('units','centimeters','position',[12 4 12.5 14])
subplot(1,1,1)
imshow(uint8(Negative_Tested_Im_Subt_Mean))
if J1<=L1

   title({['Tested Image No.' num2str(J1) .........
       ' (Mr. Mansour Alshammari, ',
       '1^{st} Projection)';
   'After Subtracting the Average Training Face';
   '(A Negative Image)'}

elseif J1>L1 && J1<=L2

   title({['Tested Image No.' num2str(J1) .........
       ' (Mr. Mansour Alshammari, ',
       '2^{nd} Projection)';
   'After Subtracting the Average Training Face';
   '(A Negative Image)'}

elseif J1>L2 && J1<=L3

   title({['Tested Image No.' num2str(J1) .........
       ' (Mr. Mansour Alshammari, ',
       '3^{rd} Projection)';
   'After Subtracting the Average Training Face';
   '(A Negative Image)'}

elseif J1>L3 && J1<=L4

   title({['Tested Image No.' num2str(J1) .........
       ' (Mr. Methkir Alharthee, ',
       '1^{st} Projection)';
   'After Subtracting the Average Training Face';
   '(A Negative Image)'}

elseif J1>L4 && J1<=L5
% title({'Tested Image No.' num2str(J1) ..... ' (Mr. Methkir Alharthee, '........
% '2^{nd} Projection)'});......
% 'After Subtracting the Average Training Face';......
% '(A Negative Image)')
elseif J1>L5 && J1<=L6
% title({'Tested Image No.' num2str(J1) ..... ' (Mr. Methkir Alharthee, '........
% '3^{rd} Projection)'});......
% 'After Subtracting the Average Training Face';......
% '(A Negative Image)')
elseif J1>L6 && J1<=L7
% title({'Tested Image No.' num2str(J1) ..... ' (Mr. Mohammed Hanafy, '........
% '1^{st} Projection)'});......
% 'After Subtracting the Average Training Face';......
% '(A Negative Image)')
elseif J1>L7 && J1<=L8
% title({'Tested Image No.' num2str(J1) ..... ' (Mr. Mohammed Hanafy, '........
% '2^{nd}Projection)'});......
% 'After Subtracting the Average Training Face';......
% '(A Negative Image)')
elseif J1>L8 && J1<=L9
% title({'Tested Image No.' num2str(J1) ..... ' (Mr. Mohammed Hanafy, '........
% '3^{rd} Projection)'});......
% 'After Subtracting the Average Training Face';......
% '(A Negative Image)')
end
% disp({'Please, press any keyboard button to explore '.....
% 'the remaining tested images'})
% disp({'after subtracting the average training face '.....
% 'from them >>>>>>>'})
% pause
% close all
% clc
end
Tested_Im_Subt_Mean;

% The calculation of a covariance matrix for all training faces.
Cov_Matrix=Known_Im_Subt_Mean*......
    transpose(Known_Im_Subt_Mean); % The covariance matrix
    % for all training faces.

% Another way for calculating the covariance matrix for all
% training faces.
    m=zeros(N*N,N*N);
    for i=1:Total_No_of_Known_Im
% C = Known_Im_Subt_Mean(:, i) * transpose(Known_Im_Subt_Mean(:, i));
% m = m + C;
% end
% Cov_Matrix = (1/Total_No_of_Known_Im) * m; % The covariance matrix
% for all training faces.

% The calculation of eigenvalues and eigenvectors for the covariance matrix.
[Eigenvectors Eigenvalues] = eig(Cov_Matrix); % Note that, The calculated covariance matrix is usually too big which makes the calculation of eigenvalues and eigenvectors very difficult if not impossible. So, it is not practical to calculate the eigenvalues and eigenvectors for the such matrix. The dimensions of the covariance matrix can be reduced to the number of the training faces as will be proved shortly.
Eigenvalues = (diag(Eigenvalues)).';

% Eigenvectors must be positive because the covariance matrix is positive definite. Due to round-off error, MATLAB makes the smallest eigenvectors negative. Then those negative eigenvectors must be set to zero.
for jjj = 1:length(Eigenvalues)
    if Eigenvalues(1, jjj) < 0
        Eigenvalues(1, jjj) = 0;
    end
end

% The calculation of a more practical covariance matrix.
New_Cov_Matrix = transpose(Known_Im_Subt_Mean) * Known_Im_Subt_Mean;

[Eigvect Eigval] = eig(New_Cov_Matrix); % The calculation of eigenvalues and eigenvectors for the reduced covariance matrix.

Eigval_Reduced_Cov = (diag(Eigval)).';
Eigvect_Reduced_Cov = Known_Im_Subt_Mean * Eigvect; % The columns of this matrix represent unnormalized eigenvectors that are calculated based on the more practical covariance matrix.

% Ordering the calculated eigenvalues from the reduced covariance matrix along with their eigenvectors in descending order as well
% as normalizing the eigenvectors.
Eigenvalues_Reduced_Cov=sort(Eigval_Reduced_Cov,'descend');
Eigenvectors_Reduced_Cov=zeros(size(Eigvect_Reduced_Cov,1),size(Eigvect_Reduced_Cov,2));
for ii=1:length(Eigenvalues_Reduced_Cov)
  for pp=1:length(Eigval_Reduced_Cov)
    if Eigenvalues_Reduced_Cov(1,ii)==Eigval_Reduced_Cov(1,pp)
      Eigenvectors_Reduced_Cov(:,ii)=Eigvect_Reduced_Cov(:,pp)/norm(Eigvect_Reduced_Cov(:,pp));
      break
    end
  end
end

% PCA and IPCA algorithms for calculating a feature vector matrix.
% A feature vector matrix is a matrix that is composed of a couple
% of eigenvectors that follow the most significant patterns of the
% correlated faces. These eigenvectors are called eigenfaces. In
% fact, the eigenvalues associated with those eigenfaces are
% corresponding to the biggest calculated eigenvalues. Note that,
% the PCA is approximately similar to the IPCA. The main difference
% between them is that the way of selecting eigenvectors which form
% the feature vector matrix.

fid=fopen('PCA vs. IPCA.txt','w'); % A text file for typing the
% required results to select the
% desired eigenvectors for the
% feature vector matrix.
fprintf(fid,['\n ***** The Results of PCA and IPCA Used to Select the Desired Eigenvectors *****
''Select the Desired Eigenvalues PIF IR(%%) AIR(%%) VCR(%%) TVC(%%)
''and Setting up Thresholds *****

'\t Transformed\n''No. Eigenvalues PIF IR(%%) AIR(%%) VCR(%%) TVC(%%)
''The PCA and IPCA Code for Testing ']
fprintf(fid,'t *** The PCA and IPCA Code for Testing ');
fprintf(fid,'
\n''The calculation of the transformed eigenvalues.
ro=zeros(1,length(Eigenvalues_Reduced_Cov));
for v=1:length(Eigenvalues_Reduced_Cov)
  ro(v,1)=1-Eigenvalues_Reduced_Cov(1,v)/sum(Eigenvalues_Reduced_Cov);
end

% The calculation of the possibility information function (PIF).
PIF=zeros(1,length(Eigenvalues_Reduced_Cov));
for nn=1:length(Eigenvalues_Reduced_Cov)
    PIF(1,nn)=-log2(ro(1,nn));
end

% The calculation of the possibility information entropy (PIE).
PIE=0;
for n1=1:length(Eigenvalues_Reduced_Cov)
    PIE=PIE-ro(1,n1)*log2(ro(1,n1));
end
PIE;

% The calculation of the information rate (IR) and accumulated
% information rate (AIR).
IR=zeros(1,length(Eigenvalues_Reduced_Cov));
AIR=zeros(1,length(Eigenvalues_Reduced_Cov));
PIF1=0;
for u=1:length(Eigenvalues_Reduced_Cov)
    IR(1,u)=PIF(1,u)/sum(PIF);
    PIF1=PIF1+PIF1;
    AIR(1,u)=PIF1/sum(PIF);
end

% The calculation of the variance contribution rate (VCR) and total
% variance contribution rate (TVC).
VCR=zeros(1,length(Eigenvalues_Reduced_Cov));
TVC=zeros(1,length(Eigenvalues_Reduced_Cov));
TVC1=0;
for Q=1:length(Eigenvalues_Reduced_Cov)
    VCR(1,Q)=Eigenvalues_Reduced_Cov(1,Q)/sum(Eigenvalues_Reduced_Cov);
    TVC1=Eigenvalues_Reduced_Cov(1,Q)+TVC1;
    TVC(1,Q)=TVC1/sum(Eigenvalues_Reduced_Cov);
    fprintf(fid,['%4.0f %12.4f %6.4f %7.4f %7.4f %8.4f %7.4f ....
    
    Eigenvalues_Reduced_Cov(1,Q),ro(1,Q),PIF(1,Q),....
    IR(1,Q)*100,AIR(1,Q)*100,VCR(1,Q)*100,TVC(1,Q)*100);
end

fprintf(fid,['==============================================' ....
'=======================================
']);
fprintf(fid,['** Notice that, in IPCA we can take the first ..... 
'seven eigenvectors associated with the\r\n']);
fprintf(fid,['** Notice that, in IPCA we can take the first ..... 
'vector matrix but in PCA we must take the\r\n']);
fprintf(fid,['First eight eigenvectors. That because when ..... 
'm=7, the AIR=95.6891 which is good\r\n']);
fprintf(fid,['If we want to pick the eigenvectors\r\n']);
fprintf(fid,['based on the TVC then we have to take the ..... 
'}
Appendix A. A Code for the Digital Model

113 'first eight eigenvectors in order to\n\n')
1377 fprintf(fid,[' ' make sure that the TVC is big enough. ' ....
1378 'Therefore, the AIR tells us more about the\n\n')
1379 fprintf(fid,[' ' information contained in the eigenfaces. ' ....
1380 '\n\n')
1381 fclose(fid);
1382
1383 disp(['Please see the documented results of PCA and IPCA in '....
1384 'the open text file. Then select the'])
1385 disp(['number of the eigenvectors for each algorithm required '....
1386 'to form the feature matrix.'])
1387 disp(['After that, press any keyboard button to resume the '.....
1388 'code >>>>>>>>>>>>')
1389 Text='PCA vs. IPCA.txt';
1390 open(Text) % Opening the text file which contains
1391 % the PCA and IPCA results.
1392 pause
1393 clc
1394 open('PCA_IPCA_Testing_and_Setting_up_Thresholds.m')
1395
1396 % Customizing and fixing the number of eigenfaces for PCA and IPCA.
1397 Prompt={[['1 to m, enter the value of m for PCA, where m is '.....
1398 'the lower bound of eigenfaces. It can take up to ' ....
1399 num2str(length(Eigenvalues_Reduced_Cov)) ':'],.....
1400 ['1 to m, enter the value of m for IPCA, where m is the '.....
1401 'lower bound of eigenfaces. It can take up to ' ....
1402 num2str(length(Eigenvalues_Reduced_Cov)) ':'],};
1403 Dlg_Title='The Lower Bound of Eigenfaces for PCA and IPCA';
1404 Num_Lines=1;
1405 Def={'8','7'};
1406 L=inputdlg(Prompt,Dlg_Title,Num_Lines,Def,'on');
1407 V=str2num(char(L));
1408 for xx=1:2
1409     if xx==1
1410         if V(xx)<1 || V(xx)>length(Eigenvalues_Reduced_Cov) || ....
1411             mod(V(xx),1)~=0 || V(xx)<=V(2)
1412             if V(xx)<1
1413                 W=errordlg(['m for PCA Is Invalid Because It '.....
1414                     'Less Than One'],'An Error Dialog');
1415             else 
1416                 W=errordlg(['m for PCA Is Invalid Because It '.....
1417                     'Is Bigger Than the Upper Bound of '.....
1418                     'Eigenfaces'],'An Error Dialog');
1419             return
1420         elseif V(xx)>length(Eigenvalues_Reduced_Cov)
1421             W=errordlg(['m for PCA Is Invalid Because It '.....
1422                     'Is Not an Integer'],'An Error Dialog');
1423             return
1424         elseif mod(V(xx),1)~=0
1425             W=errordlg(['m for PCA Is Invalid Because It '.....
1426                     'Is Not an Integer'],'An Error Dialog');
1427             return
1428         elseif V(xx)<=V(2)
1429             inputdlg(['m for IPCA Is Invalid Because It '.....
1430                     'Less Than One'],'An Error Dialog');
1431             return
1432         else 
1433             inputdlg(['m for IPCA Is Invalid Because It '.....
1434                     'Is Bigger Than the Upper Bound of '.....
1435                     'Eigenfaces'],'An Error Dialog');
1436             return
1437         end
1438     end
1439     if xx==2
1440         if V(xx)<1 || V(xx)>length(Eigenvalues_Reduced_Cov) || ....
1441             mod(V(xx),1)~=0 || V(xx)<=V(2)
1442             if V(xx)<1
1443                 W=errordlg(['m for IPCA Is Invalid Because It '.....
1444                     'Less Than One'],'An Error Dialog');
1445             else 
1446                 W=errordlg(['m for IPCA Is Invalid Because It '.....
1447                     'Is Bigger Than the Upper Bound of '.....
1448                     'Eigenfaces'],'An Error Dialog');
1449             return
1450         elseif V(xx)>length(Eigenvalues_Reduced_Cov)
1451             W=errordlg(['m for IPCA Is Invalid Because It '.....
1452                     'Is Not an Integer'],'An Error Dialog');
1453             return
1454         elseif mod(V(xx),1)~=0
1455             W=errordlg(['m for IPCA Is Invalid Because It '.....
1456                     'Is Not an Integer'],'An Error Dialog');
1457             return
1458         elseif V(xx)<=V(2)
1459             inputdlg(['m for IPCA Is Invalid Because It '.....
1460                     'Less Than One'],'An Error Dialog');
1461             return
1462         end
1463     end
1464     V=V(xx)
1465 end
W=errordlg(['m for PCA Is Invalid Because It ' ....
'Must Be Bigger Than IPCA']);
return
end
else
    if V(xx)<1 || mod(V(xx),1)~=0
        if V(xx)<1
            W=errordlg(['m for IPCA Is Invalid Because It ' ....
'Is Less Than One'], 'An Error Dialog');
            return
        elseif mod(V(xx),1)~=0
            W=errordlg(['m for IPCA Is Invalid Because It ' ....
'Is Not an Integer'], 'An Error Dialog');
            return
        end
    end
end

Best_Eigenvalues_PCA=Eigenvalues_Reduced_Cov(1,1:V(1));
Best_Eigenvectors_PCA=........

Best_Eigenvalues_IPCA=Eigenvalues_Reduced_Cov(1,1:V(2));
Best_Eigenvectors_IPCA=........

% Plotting the eigenvalues for each explained algorithm compared
% with the calculated eigenvalues from the covariance matrix for
% all training faces.
Threshold=0.1; % This threshold is for picking
% up the biggest eigenvalues.
[WW LL]=find(Eigenvalues>Threshold);
Flipped_Eigenvalues_Reduced_Cov=fliplr(Eigenvalues_Reduced_Cov);
if length(LL)<length(Flipped_Eigenvalues_Reduced_Cov)
    LL1=LL;
    LL=[LL(1)−ones(1,length(Flipped_Eigenvalues_Reduced_Cov))−....
        length(LL)) LL];
end
figure('units','centimeters','position',[6 1.2 25 16.9])
subplot(1,1,1)
The code snippet from the document is as follows:

```matlab
bar(LL, [zeros(length(LL)−length(LL1)) Eigenvalues(LL1)], 0.8, 'FaceColor', 'k', 'EdgeColor', 'k')
hold on
bar(LL, [zeros(1,length(LL)−length(Best_Eigenvalues_PCA))]
      flipr(Best_Eigenvalues_PCA)], 0.8/2, 'FaceColor', 'r', 'EdgeColor', 'r')
hold on
bar(LL, [zeros(1,length(LL)−length(Best_Eigenvalues_IPCA))]
      flipr(Best_Eigenvalues_IPCA)], 0.8/4,
      'FaceColor', 'b', 'EdgeColor', 'b')
hold off
title({['The Calculated Eigenvalues for Each Proposed ' ....
      'Algorithm Compared '];['with the Calculated Eigenvalues ' ....
      'form the Covariance Matrix for All Training Faces']})
xlabel('Eigenvalue Index')
ylabel('Eigenvalue')
legend({['The Calculated Eigenvalues from the Covariance ' ....
      'Matrix for All Training Faces.'],
      ['The Calculated ' ....
      'Eigenvalues by Using PCA Algorithm.'],
      ['The Calculated ' ....
      'Eigenvalues by Using IPCA Algorithm.'],
      'Location', 'NorthWest')
disp('Please, press any keyboard button to resume the code >>>>')
pause
clc
close all

% Displaying the calculated eigenfaces from the covariance matrix
% for all training faces.
for I=1:length(LL1)
    Eigenvalues1=flipr(LL1);
    Eigenvectors1=......
    flipdim(Eigenvectors(:,LL1),2); % This flipping just
    % to make the highest
    % correlated eigenface
    % is associated with the
    % first eigenvalue and
    % so on. This is done to
    % be compatible with the
    % generated document
    % "PCA vs. IPCA".
    X=floor(255*(double(Eigenvectors1(:,I))/......
      max(max(Eigenvectors1(:,I)))); % Normalized eigenface.
    % This is done to increase
    % the dynamic range of the
    % eigenface for
    % visualization by scaling
    % the intensities of the
    % eigenface from 0 to 255.
    Eigenface=reshape(X,N,N);
    Negative_Eigenface=255*ones(N,N)−255*....
```
(Eigenface/max(max(Eigenface))); % Obtaining a negative
% image for the eigenface
% in order to enhance its
% appearance.

figure('units','centimeters','position',[15.5 5.5 9 11.5])
subplot(1,1,1)
imshow(uint8(Negative_Eigenface))
title({['Eigenface No.' num2str(I) ............
    'Associated with Eigenvalue No.'........
    num2str(Eigenvalues1(I)) '.'];.......[
    'It Is Calculated from the '........
    'Covariance Matrix'];'for All Training Faces.';.......[
    '(A Negative Image')]})

disp(['Please, press any keyboard button to explore '......
    'the remaining calculated eigenfaces'])
disp(['from the covariance matrix for all training '......
    'faces >>>>>>>'])
pause
clc
close all
end

% Displaying the calculated eigenfaces by using PCA algorithm.
for I=1:length(Best_Eigenvalues_PCA)
    X=floor(255*(double(Best_Eigenvectors_PCA(:,I))/max(max(Best_Eigenvectors_PCA(:,I))))); % Normalized eigenface.
    % This is done to increase
    % the dynamic range of the
    % eigenface for
    % visualization by scaling
    % the intensities of the
    % eigenface from 0 to 255.
    Eigenface=reshape(X,N,N);
    Negative_Eigenface=255*ones(N,N)−255*....
    (Eigenface/max(max(Eigenface))); % Obtaining a negative
    % image for the eigenface
    % in order to enhance its
    % appearance.
    figure('units','centimeters','position',[15.5 5.5 8 10.5])
    subplot(1,1,1)
    imshow(uint8(Negative_Eigenface))
title({['Calculated Eigenface No.' num2str(I) .......
    'Associated'];['with Eigenvalue No.' num2str(I) ......
    ' by Using'];'PCA Algorithm';'(A Negative Image')})
disp(['Please, press any keyboard button to explore the '......
    'remaining calculated'])
disp('eigenfaces by using PCA algorithm >>>>>>>')
pause
clc
close all
end

% Displaying the calculated eigenfaces by using IPCA algorithm.
for I=1:length(Best_Eigenvalues_IPCA)
    X=floor(255*(double(Best_Eigenvectors_IPCA(:,I))/max(max(Best_Eigenvectors_IPCA(:,I)))));
    % Normalized eigenface.
    % This is done to increase
    % the dynamic range of the
    % eigenface for
    % visualization by scaling
    % the intensities of the
    % eigenface from 0 to 255.
    Eigenface=reshape(X,N,N);
    Negative_Eigenface=255*ones(N,N)-255*(Eigenface/max(max(Eigenface)));
    % Obtaining a negative
    % image for the eigenface
    % in order to enhance its
    % appearance.
    figure('units','centimeters','position',[15.5 5.5 8 10.5])
    subplot(1,1,1)
    imshow(uint8(Negative_Eigenface))
    title({['Calculated Eigenface No.' num2str(I)]
        [' Associated'];['with Eigenvalue No.' num2str(I)]
        [' by Using'];'IPCA Algorithm';'(A Negative Image)'}
    disp(['Please, press any keyboard button to explore the ']
        ['remaining calculated'])
    disp('eigenfaces by using IPCA algorithm >>>>>>>')
    pause
clc
close all
end

save('Eigenvectors','Eigenvectors')
save('Best_Eigenvectors_PCA','Best_Eigenvectors_PCA')
save('Best_Eigenvectors_IPCA','Best_Eigenvectors_IPCA')
break

%%
%% The Compression and Reconstruction of the Training Faces %%%
close all

When a small number of eigenfaces is used to compress and reconstruct the training faces then the processing speed will increase so the IPCA algorithm is the fastest one then PCA algorithm finally the smallest processing speed occurs when all calculated eigenvectors from the covariance matrix for all training faces are used as eigenfaces. When a small number of the eigenfaces is used to project and reconstruct the training faces then the training faces will have bad quality. Therefore, the highest error in reconstruction occurs when the IPCA algorithm is used; then the PCA algorithm comes second; finally, the usage of all eigenvectors as eigenfaces produces the smallest reconstruction error.

Best_Eigenvectors=Eigenvectors; % Just try it to see how affects on the reconstructed training faces.

% Best_Eigenvectors=Best_Eigenvectors_PCA; % Just try it to see how affects on the reconstructed training faces.

Best_Eigenvectors=Best_Eigenvectors_IPCA; % Just try it to see how affects on the reconstructed training faces.

% Projecting each training face on the eigenspace. It is just expressing each training face in terms of the eigenfaces. This is called principal components transform (also called the Hotelling or Karhunen–Loéve transform)

All_Known_Transformed_Im=.............. zeros(size(Best_Eigenvectors,2),........ Total_No_of_Known_Im); % A 2D matrix where each column represents the coordinates of a projected training face in the eigenspace.

for r=1:Total_No_of_Known_Im
    All_Known_Transformed_Im(:,r)=Best_Eigenvectors.'*........ Known_Im_Subt_Mean(:,r);
end
All_Known_Transformed_Im;

% The reconstruction of the projected training faces.
All_Known_Reconstructed_Im_V=.............. zeros(N*N,Total_No_ofKNOWN_Im); % An N^2xP, 2D matrix where
% each column represents a
% reconstructed training face
% vector.
All_Known_Reconstructed_Im=zeros(N,N,Total_No_of_Known_Im);
for a=1:Total_No_of_Known_Im
    Pre=reshape(Best_Eigenvectors*............
    All_Known_Transformed_Im(:,a),N,N)+............
    reshape(Av_Image,N,N); % Adding the average training face.
    RC1=Rows_Columns(:,:,a);
    F=RC1(1,:);
    F1=RC1(2,:);
    D=F(F>0);
    D1=F1(F1>0);
    RC=[D;D1]; % The rows and columns for the pixels of a face
    % after removing the added zero rows and columns.
    % The added zero rows and columns are added by % MATLAB for making the matrices of the rows and
    % columns of the faces pixels are equal.
    for QQ=1:size(RC,2)
        All_Known_Reconstructed_Im(RC(1,QQ),RC(2,QQ),a)=.......
        Pre(RC(1,QQ),RC(2,QQ))+Means(1,a); % Adding the mean
        % of the pixels of
        % a face.
    end
    Y=All_Known_Reconstructed_Im(:,:,a); % Removing any pixel less
    % than zero in a training
    % face because the image
    % can not be negative.
    [UU NN]=find(Y<0);
    for CC=1:size(UU,1)
        Y(UU(CC),NN(CC))=0;
    end
    All_Known_Reconstructed_Im_V(:,a)=reshape(Y,N*N,1);
end
Known_Image=Known_Images(a).name;
figure('units','centimeters','position',[16 7 7.5 8.5])
subplot(1,1,1)
imshow(uint8(reshape(All_Known_Reconstructed_Im_V(:,a),N,N)))
if a==1
    title({[['Reconstructed Training Face No.' num2str(a)];....
            ['(', 'Known_Image(1:length(Known_Image)-6) ........
            ', 1^(st) Projection')]])
elseif a==2
    title({[['Reconstructed Training Face No.' num2str(a)];....
            ['(', 'Known_Image(1:length(Known_Image)-6) ........
            ', 2^(nd) Projection')]])
else a==3
title({'Reconstructed Training Face No.' num2str(a)});
elseif a==4
    title({'Reconstructed Training Face No.' num2str(a)});
elseif a==5
    title({'Reconstructed Training Face No.' num2str(a)});
elseif a==6
    title({'Reconstructed Training Face No.' num2str(a)});
elseif a==7
    title({'Reconstructed Training Face No.' num2str(a)});
elseif a==8
    title({'Reconstructed Training Face No.' num2str(a)});
elseif a==9
    title({'Reconstructed Training Face No.' num2str(a)});
end
disp('Please, press any keyboard button to explore ');
disp('the remaining')
disp('reconstructed training faces >>>>>>>')
pause
close all
clc
All_Known_Reconstructed_Im_V;

%%
% The Calculation of Compression and Reconstruction Performance
clc
close all

After_Compression_Normalization=zeros(1,length(Eigenvalues));
% Measuring an information rate after compression. The elements of this
% vector represent the information
% rates after compression with respect
% to the information rates before
% compression.
MSE_Compression=....... zeros(1,length(Eigenvalues)); % Measuring compression
% performance or how much
% compressed information is.
% The elements of this vector
% represent the mean squared
% errors in compression when
% different eigenfaces are
% selected.
MSE_Reconstruction=zeros(length(Eigenvalues),..........
Total_No_of_Known_Im); % Measuring reconstruction performance
% or the quality of a reconstructed
% training face. The elements of each
% column of this matrix represent the
% mean squared errors between a
% projected training face and its
% reconstruction when different
% eigenfaces are selected.
Eigenvalues1=fliplr(Eigenvalues);
Eigenvectors1=flipdim(Eigenvectors,2); % This flipping just to make
% the highest correlated
% eigenface is associated
% with the first eigenvalue
% and so on. This is done to
% be compatable with the
% generated document
% "PCA vs. IPCA".
for kk=1:length(Eigenvalues)
    ['Iteration No.: ' num2str(kk) ' Out of ' ............
    num2str(length(Eigenvalues))]
Selected_Eigenvectors=Eigenvectors1(:,1:kk);

% Projecting each training face on the eigenspace. It is just
% expressing each training face in terms of the eigenfaces.
% This is called principal components transform (also called
% the Hotelling or Karhunen–Loève transform)
All_Known_Transformed_Im=.............. zeros(size(Selected_Eigenvectors,2),........
Total_No_of_Known_Im); % A 2D matrix where each column
% represents the coordinates of a
% projected training face in the
% eigenspace.
for r=1:Total_No_of_Known_Im
    All_Known_Transformed_Im(:,r)=Selected_Eigenvectors.'*......
    Known_Im_Subt_Mean(:,r);
end
All_Known_Transformed_Im;
% The calculation of an information ratio before and after % compression.

Before_Compression=Total_No_of_Known_Im*....
N*N;  % The overall information when there is no any compression technique is used.

After_Compression=N*N*kk+kk*Total_No_of_Known_Im+............
N*N;  % The overall information when a compression technique is used. The overall information here is controlled % by the selected number of eigenvectors. When the % selected number is small then the information will % be small and vice versa. It is very important to % notice that when all eigenvectors are used then % there will not be any compression and the overall % information when there is no any compression % technique is used will be the optimum one. Also, it % is really important to notice that the rows and % columns for the pixels of each face as well as the % means of the faces pixels are not added to the % overall information after compression that because % the face centering operation is not really important % for the image compression process and will not have % any effect if it is done or not but it has been done % here because it is important for other processes.

Before_Compression_Normalization=(Before_Compression/....... Before_Compression)*100;  % Note that, the normalization is done to make the overall % information before compression % is equal to 100 all the time % in order to make comparison % easier.

After_Compression_Normalization(1,kk)=(After_Compression/..... Before_Compression)*100;  % Note that, the normalization is done to make the overall % information before compression % is equal to 100 all the time % in order to make comparison % easier.

The_Information_Ratio=[..............
% num2str(Before_Compression_Normalization) ............
% (Before Compression) : ' ..............
% num2str(After_Compression_Normalization(1,kk))....
% (After Compression)']  % This is just to make the % information ratio is readable % on the command window.

% The reconstruction of the projected training faces.
All_Known_Reconstructed_Im=zeros(N,N,Total_No_of_Known_Im);
for a=1:Total_No_of_Known_Im
  Pre=reshape(Selected_Eigenvectors*............
Appendix A. A Code for the Digital Model

All_Known_Transformed_Im(:,a),N,N)+..........
reshape(Av_Image,N,N); % Adding the average
% training face.

RC1=Rows_Columns(:,a);
F=RC1(1,:);
F1=RC1(2,:);
D=F(F>0);
D1=F1(F1>0);
RC=[D;D1]; % The rows and columns for the pixels of a
% face after removing the added zero rows and
% columns. The added zero rows and columns are
% added by MATLAB for making the matrices of
% the rows and columns of the faces pixels
% are equal.
for QQ=1:size(RC,2)
    All_Known_Reconstructed_Im(RC(1,QQ),RC(2,QQ),a)=..........
% Pre(RC(1,QQ),RC(2,QQ))+..........
    Means(1,a); % Adding the mean of the
    % pixels of a face.
end

Y=All_Known_Reconstructed_Im(:,a); % Removing any pixel
% less than zero in
% a training face
% because the image
% can not be negative.
[UU NN]=find(Y<0);
for CC=1:size(UU,1)
    Y(UU(CC),NN(CC))=0;
end

MSE_Compression(1,kk)=sum(Eigenvalues1(1,kk+1:end));
MSE_Reconstruction(kk,a)=..........
    sum(sum(...........
        (reshape(Normalized_Known_Im_V(:,a),N,N)-Y).^2))/N*N;

% % Displaying the effect of the selected eigenvectors on
% % the reconstructed training faces.
% Known_Image=Known_Images(a).name;
% figure('units','centimeters','position',[16 7 7.5 8.5])
% subplot(3,1,1)
% imshow(uint8(reshape(Normalized_Known_Im_V(:,a),N,N)))
if a==1
    title({['Original Training Face No.' num2str(a)];....
        [' Known_Image(1:length(Known_Image)-6) ..... 
        ', 1^(st) Projection']})
elseif a==2
    title({['Original Training Face No.' num2str(a)];....
        [' Known_Image(1:length(Known_Image)-6) ..... 
        ', 1^(st) Projection']})
elseif a==3
    title({'Original Training Face No.' num2str(a);....
          [' Known Image(1:length(Known Image)-6) ......
          ', 2^{nd} Projection]')}
elseif a==4
    title({'Original Training Face No.' num2str(a);....
          [' Known Image(1:length(Known Image)-6) ......
          ', 3^{rd} Projection]'})
elseif a==5
    title({'Original Training Face No.' num2str(a);....
          [' Known Image(1:length(Known Image)-6) ......
          ', 1^{st} Projection]'})
elseif a==6
    title({'Original Training Face No.' num2str(a);....
          [' Known Image(1:length(Known Image)-6) ......
          ', 2^{nd} Projection]'})
elseif a==7
    title({'Original Training Face No.' num2str(a);....
          [' Known Image(1:length(Known Image)-6) ......
          ', 1^{st} Projection]'})
elseif a==8
    title({'Original Training Face No.' num2str(a);....
          [' Known Image(1:length(Known Image)-6) ......
          ', 2^{nd} Projection]'})
elseif a==9
    title({'Original Training Face No.' num2str(a);....
          [' Known Image(1:length(Known Image)-6) ......
          ', 3^{rd} Projection]'}
end
subplot(3,1,2)
imshow(uint8(Y))
if a==1
    title({'Its Reconstruction When q=' num2str(kk)})
elseif a==2
    title({'Its Reconstruction When q=' num2str(kk)})
elseif a==3
    title({'Its Reconstruction When q=' num2str(kk)})
elseif a==4
    title({'Its Reconstruction When q=' num2str(kk)})
elseif a==5
    title({'Its Reconstruction When q=' num2str(kk)})
elseif a==6
    title({'Its Reconstruction When q=' num2str(kk)})
elseif a==7
    title({'Its Reconstruction When q=' num2str(kk)})
elseif a==8
    title({'Its Reconstruction When q=' num2str(kk)})
elseif a==9
    title({'Its Reconstruction When q=' num2str(kk)})
else
    title({'Its Reconstruction When q=' num2str(kk)})
% end
% subplot(3,1,3)
% imshow(uint8(..........
% (reshape(Normalized_Known_Im_V(:,a),N,N)-Y).^2))
if a==1
    title('The Squared Error')
elseif a==2
    title('The Squared Error')
elseif a==3
    title('The Squared Error')
elseif a==4
    title('The Squared Error')
elseif a==5
    title('The Squared Error')
elseif a==6
    title('The Squared Error')
elseif a==7
    title('The Squared Error')
elseif a==8
    title('The Squared Error')
elseif a==9
    title('The Squared Error')
end
% kk
% pause
% close all
end
end

% Plotting the information rates when different eigenfaces are
% selected compared with the original information rate for all
% training faces and explaining the information rates for the PCA
% and IPCA algorithms on the plot.
clc
figure('units','centimeters','position',[0.15 1.2 35.8 16.9])
subplot(2,1,1)
Leg1=plot(1:length(Eigenvalues),After_Compression_Normalization);
Leg2=line([0 length(Eigenvalues)],[100 100],'Color',[0 102/255 0]);
hold on
Leg3=plot(1,After_Compression_Normalization(1),'kd',.....
    'LineWidth',1.5,'MarkerEdgeColor','k','MarkerFaceColor',.....
    'm','MarkerSize',8);
text(-50,After_Compression_Normalization(1)+3650,.....
    {'\fontsize{10} \color{black}' .......
    After_Compression_Normalization(1))
hold on
Leg4=plot(length(Eigenvalues),........
    After_Compression_Normalization(length(Eigenvalues)),'kd',.....
Appendix A. A Code for the Digital Model

```
'LineWidth',1.5,'MarkerEdgeColor','k','MarkerFaceColor',.....
'g','MarkerSize',8);

After_Compression_Normalization=length(Eigenvalues))+2350,.....
'\fontsize{10} \color{black}'.........
After_Compression_Normalization=length(Eigenvalues)))
hold off
axis([-100 length(Eigenvalues)+(length(Eigenvalues)/40) -2000 ..... 
max(After_Compression_Normalization)+.........
(max(After_Compression_Normalization)/10)])
set(gca,'XTick',[1 length(Eigenvalues)/2 length(Eigenvalues)])
title(['The Information Rates When Different Eigenfaces '.....
'Are Selected compared to '];........
'With the Original Information Rate for All Training Faces'})
xlabel('The Number of Selected Eigenfaces')
ylabel('The Information Rate (%)')
legend([Leg1 Leg2 Leg3 Leg4],[['The Information Rates When '.....
'Different Eigenfaces Are Selected.'],[['The Information '.....
'Rate for All Training Faces.'],[['The Information Rate '.....
'When the First Eigenface Is Selected.'],[['The '.....
'Information Rate When All Eigenfaces Are Selected.']]])

'Location','NorthWest');

subplot(2,1,2)
Leg1=plot(1:length(Eigenvalues),After_Compression_Normalization);
axis([0 20 0 200])
Leg2=line([0 length(Eigenvalues)],[100 100],'Color',[0 102/255 0]);
Leg3=line([length(Best_Eigenvalues_PCA) length(Eigenvalues)])
axis([0 20 0 200])
Leg4=line([length(Best_Eigenvalues_PCA) length(Eigenvalues)])

num2str(After_Compression_Normalization('%6.4f'))
```

After Compression Normalization(......
length(Best_Eigenvalues_IPCA)),'LineStyle','--','Color','r','LineWidth',4)
text(0.08,After Compression Normalization(......
length(Best_Eigenvalues_IPCA))-1.7,......
{'\fontsize{10} \color{red} \bf' ......
num2str(After Compression Normalization(......
length(Best_Eigenvalues_IPCA)),'%6.4f'))
vv=[length(Best_Eigenvalues_IPCA) length(Best_Eigenvalues_PCA)];
if vv(1)==1 || vv(2)==1
set(gca,'XTick',[0 sort(vv) 20])
else set(gca,'XTick',[0 1 sort(vv) 20])
end
title(['Explaining the Information Rates for the PCA and '......
'IPCA Algorithms on the Plot'])
xlabel('The Number of Selected Eigenfaces')
ylabel('The Information Rate (%)')
legend([Leg1 Leg2 Leg3 Leg4],['The Information Rates When '......
'Different Eigenfaces Are Selected.'],['The Information Rate '......
'When PCA Algorithm Is Used.'],['The Information '......
'Rate When IPCA Algorithm Is Used.'],'Location','SouthEast');
disp('Please, press any keyboard button to resume the code >>>>)
pause
clc
close all

% Plotting the mean squared errors of compression for different
% selected eigenfaces compared with the mean squared errors for
% the PCA and IPCA algorithms.
figure('units','centimeters','position',[0.15 1.2 35.8 16.9])
subplot(2,1,1)
Leg1=plot(1:length(Eigenvalues),MSE_Compression);
hold on
Leg2=plot(1,MSE_Compression(1),'kd','LineWidth',1.5,......
'MarkerEdgeColor','k','MarkerFaceColor','m','MarkerSize',8);
text(25,MSE_Compression(1)+(10^7/11),......
{'\fontsize{10} \color{black}' MSE_Compression(1)})
hold on
Leg3=plot(length(Eigenvalues),......
MSE_Compression(length(Eigenvalues)),'kd','LineWidth',1.5,......
'MarkerEdgeColor','k','MarkerFaceColor','g','MarkerSize',8);
text(length(Eigenvalues)-8,......
MSE_Compression(length(Eigenvalues))+10^7/3.2,......
{'\fontsize{10} \color{black}' MSE_Compression(length(Eigenvalues))})
hold off
axis([0 length(Eigenvalues)+(length(Eigenvalues)/40) ......
-max(MSE_Compression)/10 ......
max(MSE_Compression)+(max(MSE_Compression)/10))
set(gca,'XTick',[1 length(Eigenvalues)/2 length(Eigenvalues)])
title(["The Mean Squared Errors of Compression for Different '....
'Selected Eigenfaces']
xlabel('The Number of Selected Eigenfaces')
ylabel('The Mean Squared Error')
legend([Leg1 Leg2 Leg3],['The Mean Squared Errors of '......
'Compression for Different Selected Eigenfaces.',.....
'The Mean Squared Error of Compression When the First '.....
'Eigenface Is Selected.',['The Mean Squared Error of '.....
'Compression When All Eigenfaces Are Selected.'],........
'Location','NorthEast');

subplot(2,1,2)
Leg1=plot(1:length(Eigenvalues),MSE_Compression);
axis([0 20 -10^7/4 max(MSE_Compression)+(10^7/4)])
Leg2=line([length(Best_Eigenvalues_PCA) ......
length(Best_Eigenvalues_PCA)],........
[0 MSE_Compression(length(Best_Eigenvalues_PCA))],.....
'LineStyle','--','Color','k','LineWidth',3);
line([0 length(Best_Eigenvalues_PCA)],........
[MSE_Compression(length(Best_Eigenvalues_PCA)) ......
'MSE_Compression(length(Best_Eigenvalues_PCA))],.....
'LineStyle','--','Color','k','LineWidth',3)
text(0.08,MSE_Compression(length(Best_Eigenvalues_PCA))+......
(-0.008*10^7),{"\fontsize{10} \color{black}num2str(MSE_Compression(length(Best_Eigenvalues_PCA)))})
Leg3=line([length(Best_Eigenvalues_IPCA) ......
length(Best_Eigenvalues_IPCA)],[0 ......
MSE_Compression[length(Best_Eigenvalues_IPCA)]],.....
'LineStyle','--','Color','r','LineWidth',4);
line([0 length(Best_Eigenvalues_IPCA) ......
[MSE_Compression(length(Best_Eigenvalues_IPCA)) ......
MSE_Compression(length(Best_Eigenvalues_IPCA))],.....
'LineStyle','--','Color','r','LineWidth',4)
text(0.08,MSE_Compression(length(Best_Eigenvalues_IPCA))+......
0.24*10^7,{"\fontsize{10} \color{red} \bf num2str(MSE_Compression(length(Best_Eigenvalues_IPCA)))})
if vv(1)==1 || vv(2)==1
set(gca,'XTick',[0 sort(vv) 20])
ext(gca,'XTick',[0 1 sort(vv) 20])
end
title(["Explaining the Mean Squared Errors for the PCA and '....
'IPCA Algorithms on the Plot']
xlabel('The Number of Selected Eigenfaces')
ylabel('The Mean Squared Error')
legend([Leg1 Leg2 Leg3],['The Mean Squared Errors of '.....
'Compression for Different Selected Eigenfaces.',.....
'The Mean Squared Error for PCA Algorithm.',.....
'The Mean Squared Error for IPCA Algorithm.'],.....
'Location','NorthEast');
Appendix A. A Code for the Digital Model

```matlab
disp('Please, press any keyboard button to resume the code >>>>\n')
pause
clc
close all

% Plotting the mean squared errors between each training face and
% its reconstruction for different selected eigenfaces compared
% with the mean squared errors between each training face and its
% reconstruction for the PCA and IPCA Algorithms.
figure('units','centimeters','position',[0.15 1.2 35.8 16.9])
for jj=1:Total_No_of_Known_Im
    subplot(2,1,1)
    Leg1=plot(1:length(Eigenvalues),(MSE_Reconstruction(:,jj)).');
    hold on
    Leg2=plot(1,MSE_Reconstruction(1,jj),'kd','LineWidth',1.5,....
    'MarkerEdgeColor','k','MarkerFaceColor','m',.....
    'MarkerSize',8);
    text(35,MSE_Reconstruction(1,jj)+.......
    max(MSE_Reconstruction(:,jj))/22,........
    {'\fontsize{10} \color{black}' MSE_Reconstruction(1,jj)})
    hold on
    Leg3=plot(length(Eigenvalues),.......
    MSE_Reconstruction(length(Eigenvalues),jj),'kd',.....
    'LineWidth',1.5,'MarkerEdgeColor','k',.....
    'MarkerFaceColor','g','MarkerSize',8);
    text(length(Eigenvalues)-75,.......
    MSE_Reconstruction(length(Eigenvalues),jj)+.......
    max(MSE_Reconstruction(:,jj))/7,........
    {'\fontsize{10} \color{black}' MSE_Reconstruction(length(Eigenvalues),jj)})
    hold off
    axis([0 length(Eigenvalues)+(length(Eigenvalues)/15) .....
    -max(MSE_Reconstruction(:,jj))/10 .....
    max(MSE_Reconstruction(:,jj))+........
    (max(MSE_Reconstruction(:,jj))/10)])
    set(gca,'XTick',
    Known_Image=Known_Images(jj).name;
    if jj==1
        title(['The Mean Squared Errors of Reconstructing ...'
            'Training']; ['Face No.' num2str(jj) .....'
            ' for Different Selected Eigenfaces'];....
        ['(' Known_Image(1:length(Known_Image)-6) .....'
            ', 1^(st) Projection']]}
    elseif jj==2
        title(['The Mean Squared Errors of Reconstructing ...'
            'Training']; ['Face No.' num2str(jj) .....'
            ' for Different Selected Eigenfaces'];....
        ['(' Known_Image(1:length(Known_Image)-6) .....'
            ', 2^(nd) Projection')]}
```

elseif jj==3
    title({'The Mean Squared Errors of Reconstructing ' ,'Training'; ['Face No.' num2str(jj) ' for Different Selected Eigenfaces'];
     ['(' Known_Image(1:length(Known_Image)-6) ', 3^{rd} Projection')'])
elseif jj==4
    title({'The Mean Squared Errors of Reconstructing ' ,'Training'; ['Face No.' num2str(jj) ' for Different Selected Eigenfaces'];
     ['(' Known_Image(1:length(Known_Image)-6) ', 1^{st} Projection')'])
elseif jj==5
    title({'The Mean Squared Errors of Reconstructing ' ,'Training'; ['Face No.' num2str(jj) ' for Different Selected Eigenfaces'];
     ['(' Known_Image(1:length(Known_Image)-6) ', 2^{nd} Projection')'])
elseif jj==6
    title({'The Mean Squared Errors of Reconstructing ' ,'Training'; ['Face No.' num2str(jj) ' for Different Selected Eigenfaces'];
     ['(' Known_Image(1:length(Known_Image)-6) ', 3^{rd} Projection')'])
elseif jj==7
    title({'The Mean Squared Errors of Reconstructing ' ,'Training'; ['Face No.' num2str(jj) ' for Different Selected Eigenfaces'];
     ['(' Known_Image(1:length(Known_Image)-6) ', 1^{st} Projection')'])
elseif jj==8
    title({'The Mean Squared Errors of Reconstructing ' ,'Training'; ['Face No.' num2str(jj) ' for Different Selected Eigenfaces'];
     ['(' Known_Image(1:length(Known_Image)-6) ', 2^{nd} Projection')'])
elseif jj==9
    title({'The Mean Squared Errors of Reconstructing ' ,'Training'; ['Face No.' num2str(jj) ' for Different Selected Eigenfaces'];
     ['(' Known_Image(1:length(Known_Image)-6) ', 3^{rd} Projection')'])
end
xlabel('The Number of Selected Eigenfaces')
ylabel('The Mean Squared Error')
Leg=legend([Leg1 Leg2 Leg3],['The Mean Squared Errors of ' 
'Reconstruction for Different Selected Eigenfaces.'],
['The Mean Squared Error of Reconstruction When the ' 
'First Eigenface Is Selected.'],
'The Mean Squared Error of Reconstruction When All Eigenfaces');
Appendix A. A Code for the Digital Model

Are Selected.'), 'Location', 'North');

subplot(2,1,2)
MSE_Recons=(MSE_Reconstruction(:,jj)).';
Leg1=plot(1:length(Eigenvalues),MSE_Recons);
axis([0 20 -max(MSE_Recons)/6......
max(MSE_Recons)+(max(MSE_Recons)/10)])
Leg2=line([length(Best_Eigenvalues_PCA)......
length(Best_Eigenvalues_PCA)],[0......
'MSE_Recons(length(Best_Eigenvalues_PCA))],.....
'LineStyle', '−−', 'Color', 'k', 'LineWidth', 3);
line([0 length(Best_Eigenvalues_PCA)],......
[MSE_Recons(length(Best_Eigenvalues_PCA))......
'MSE_Recons(length(Best_Eigenvalues_PCA))],......
'LineStyle', '−−', 'Color', 'k', 'LineWidth', 3)
text(0.1,MSE_Recons(length(Best_Eigenvalues_PCA))......
−(max(MSE_Recons)/88),{'ontsize{10} \color{black}'......
um2str(MSE_Recons(length(Best_Eigenvalues_PCA))))})
Leg3=line([length(Best_Eigenvalues_IPCA)......
length(Best_Eigenvalues_IPCA)],......
[0 MSE_Recons(length(Best_Eigenvalues_IPCA))],......
'LineStyle', '−−', 'Color', 'r', 'LineWidth', 4);
line([0 length(Best_Eigenvalues_IPCA)],......
[MSE_Recons(length(Best_Eigenvalues_IPCA))......
MSE_Recons(length(Best_Eigenvalues_IPCA))],......
'LineStyle', '−−', 'Color', 'r', 'LineWidth', 4)
text(0.1,MSE_Recons(length(Best_Eigenvalues_IPCA))......
+(max(MSE_Recons)/8),{'ontsize{10} \color{red} \bf'......
um2str(MSE_Recons(length(Best_Eigenvalues_IPCA))))})
if vv(1)==1 || vv(2)==1
set(gca,'XTick',[0 sort(vv) 20])
else set(gca,'XTick',[0 1 sort(vv) 20])
end
title(['Explaining the Mean Squared Errors for the '......
'PCA and IPCA Algorithms on the Plot'])
xlabel('The Number of Selected Eigenfaces')
ylabel('The Mean Squared Error')
Leg=legend([Leg1 Leg2 Leg3],[
'The Mean Squared Errors of '......
'Reconstruction for Different Selected Eigenfaces.'],.....
'The Mean Squared Error for PCA Algorithm.',.....
'The Mean Squared Error for IPCA Algorithm.',.....
'Location', 'NorthEast');
disp(['Please, press any keyboard button to explore '......
'the remaining mean'])
disp(['squared errors for other reconstructed training '......
'faces >>>>>>>'])
pause
clc
disp(['Please, press any keyboard button to explore '......
'the remaining mean'])
disp(['squared errors for other reconstructed training '......
'faces >>>>>>>'])
pause
clc
disp(['Please, press any keyboard button to explore '......
'the remaining mean'])
disp(['squared errors for other reconstructed training '......
'faces >>>>>>>'])
pause
clc
end
close all
A Face Recognition Process

```matlab
clc
close all

%% When the coordinates number of the projected training faces in the eigenspace increases, the error rate will decrease and vice versa as well as when the number decreases, the processing speed will increase and vice versa. So, the usage of all calculated eigenvectors as eigenfaces will lead to the smallest error rate and biggest processing time then the usage of the calculated eigenfaces by using PCA algorithm finally the usage of the calculated eigenfaces by using IPCA algorithm will lead to the biggest error rate and smallest processing time. It is very important to notice that the calculation of all eigenvectors from the covariance matrix for all training faces is too difficult because the covariance matrix is too big as explained before so it is impractical to use all eigenvectors as eigenfaces.

% Best_Eigenvectors=Eigenvectors; % Just try it to see how affects on the results of face recognition.
Best_Eigenvectors=Best_Eigenvectors_PCA; % Just try it to see how affects on the results of face recognition.
% Best_Eigenvectors=Best_Eigenvectors_IPCA; % Just try it to see how affects on the results of face recognition.

% Projecting each training face on the eigenspace. It is just expressing each training face in terms of the eigenfaces. This is called principal components transform (also called the Hotelling or Karhunen-Loève transform)
All_Known_Transformed_Im=....... zeros(size(Best_Eigenvectors,2),......
Total_No_of_Known_Im); % A 2D matrix where each column represents the coordinates of a projected training face in the eigenspace.
for r=1:Total_No_of_Known_Im
All_Known_Transformed_Im(:,r)=Best_Eigenvectors.'*......
Known_Im_Subt_Mean(:,r);
end
All_Known_Transformed_Im;

% Projecting each tested image on the eigenspace. It is just
% expressing each tested image in terms of the eigenfaces. This is
% called principal components transform (also called the Hotelling
% or Karhunen-Loéve transform)
All_Tested_Transformed_Im=......
zeros(size(Best_Eigenvectors,2),.....
Total_No_of_Tested_Im); % A 2D matrix where each
% column represents the
% coordinates of a
% projected tested image
% in the eigenspace.
for r1=1:Total_No_of_Tested_Im
    All_Tested_Transformed_Im(:,r1)=Best_Eigenvectors.'*......
    Tested_Im_Subt_Mean(:,r1);
end
All_Tested_Transformed_Im;

fid=fopen('Face Recognition Results for Testing.txt',.....
'w'); % A text file for typing
% the results of face
% recognition.
fprintf(fid,['
	 ***** The Results of Face Recognition '.......
'Omitted from the PCA and IPCA *****
'

'Thresholds *****

''Image No. The Image Is Originally for ' ......
'The Image Is Recognized as The Status\r\n'];
fprintf(fid,['============= =========================== 

'------------------- -------------------']);
Distances_Vector=......
zeros(Total_No_of_Tested_Im,.....
Total_No_of_Known_Im); % A 2D matrix where each row
% represents the distances
% between the weights of each
% training face and the
% weights of one of the
% tested images.
% The distances between the weights of a training face and the
% weights of each corresponding tested image. Note that, here the
% training face and tested images have the same face and projection
% so these distances must be the smallest. The distances are
% extracted form the Distances_Vector matrix.
Distances_Vector_P1=zeros(1,Im_P(1));
Distances_Vector_P2=zeros(1,Im_P(2));
Distances_Vector_P3=zeros(1,Im_P(3));
Distances_Vector_P4=zeros(1,Im_P(4));
Distances_Vector_P5=zeros(1,Im_P(5));
Distances_Vector_P6=zeros(1,Im_P(6));
Distances_Vector_P7=zeros(1,Im_P(7));
Distances_Vector_P8=zeros(1,Im_P(8));
Distances_Vector_P9=zeros(1,Im_P(9));

for p=1:Total_No_of_Tested_Im
    for q=1:Total_No_of_Known_Im
        Distances_Vector(p,q)=norm(All_Tested_Transformed_Im(:,p)−All_Known_Transformed_Im(:,q));
    end
end

if p<=L1
    Distances_Vector_P1(1,p)=Distances_Vector(p,1);
elseif p>L1 && p<=L2
    Distances_Vector_P2(1,p−L1)=Distances_Vector(p,2);
elseif p>L2 && p<=L3
    Distances_Vector_P3(1,p−L2)=Distances_Vector(p,3);
elseif p>L3 && p<=L4
    Distances_Vector_P4(1,p−L3)=Distances_Vector(p,4);
elseif p>L4 && p<=L5
    Distances_Vector_P5(1,p−L4)=Distances_Vector(p,5);
elseif p>L5 && p<=L6
    Distances_Vector_P6(1,p−L5)=Distances_Vector(p,6);
elseif p>L6 && p<=L7
    Distances_Vector_P7(1,p−L6)=Distances_Vector(p,7);
elseif p>L7 && p<=L8
    Distances_Vector_P8(1,p−L7)=Distances_Vector(p,8);
elseif p>L8 && p<=L9
    Distances_Vector_P9(1,p−L8)=Distances_Vector(p,9);
end

d1=Distances_Vector_P1;
d2=Distances_Vector_P2;
d3=Distances_Vector_P3;
d4=Distances_Vector_P4;
d5=Distances_Vector_P5;
d6=Distances_Vector_P6;
d7=Distances_Vector_P7;
d8=Distances_Vector_P8;
d9=Distances_Vector_P9;

% The calculation of the mean and standard deviation for each
% vector of the minimum distances and stacking them in a vector for
% the means and another for the standard deviations. This is done
Appendix A. A Code for the Digital Model

% for setting up a threshold for face recognition because when a
% distance between a training face and a tested image is the
% smallest with respect to the other training faces, that does not
% mean the tested image is recognized as the training face due to
% the tested image can be different than the training face and has
% the smallest distance in the same time. So, a certain threshold
% must be used to increase the accuracy of recognition.
P_Mean=[mean(d1);mean(d2);mean(d3);mean(d4);mean(d5);mean(d6);...
    mean(d7);mean(d8);mean(d9)];
P_STD=[std(d1);std(d2);std(d3);std(d4);std(d5);std(d6);std(d7);....
    std(d8);std(d9)];
% save(['Means for Face Recognition When All Eigenvectors '....
% 'Are Used as Eigenfaces'],'P_Mean')
% save(['STDs for Face Recognition When All Eigenvectors '.....
% 'Are Used as Eigenfaces'],'P_STD')
% save(['Means for Face Recognition When the Eigenfaces '....
% 'Computed by Using PCA Algorithm Are Used'],'P_Mean')
% save(['STDs for Face Recognition When the Eigenfaces '.....
% 'Computed by Using PCA Algorithm Are Used'],'P_STD')
% save(['Means for Face Recognition When the Eigenfaces '....
% 'Computed by Using IPCA Algorithm Are Used'],'P_Mean')
% save(['STDs for Face Recognition When the Eigenfaces '.....
% 'Computed by Using IPCA Algorithm Are Used'],'P_STD')

Failures_Vector=zeros(1,......
    Total_No_of_Tested_Im); % A vector for counting the number of
    % failures in the face recognition
    % process.

Latex_Matrix=cell(Total_No_of_Tested_Im,4); % This matrix is used
    % for creating a table
    % in Latex.

for w=1:Total_No_of_Tested_Im
    % The face recognition process.
    for h=1:Total_No_of_Known_Im
        if min(Distances_Vector(w,:))==........
            Distances_Vector(w,h) && ....
            min(Distances_Vector(w,:))>=......
            (P_Mean(h,1)−P_STD(h,1)) &&....
            min(Distances_Vector(w,:))<=......
            (P_Mean(h,1)+P_STD(h,1))
            s=transpose(struct2cell(Known_Images));
            c=sortrows(s,1);
            z=c(:,1);
            Recognized_As=char(z(h,1));
            Recognized_Image_Location=........
            fullfile(Known_Images_Folder,Recognized_As);
            Recognized_image=im2double(rgb2gray(.......
                imread(Recognized_Image_Location)));
            break;
else
    Recognized_As='Unknown Image';
    Recognized_image=imread('Unknown_FACE.jpg');
end

% Defining the face.
if w<=L3
    n='Mr. Mansour Alshammari';
elseif L3<w && w<=L6
    n='Mr. Methkir Alharthee';
else
    n='Mr. Mohammed Hanafy';
end

y1=strcmp(Recognized_As(1:length(Recognized_As)-6), 'Mr. Mansour Alshammari');
y2=strcmp(Recognized_As(1:length(Recognized_As)-6), 'Mr. Methkir Alharthee');
y3=strcmp(Recognized_As(1:length(Recognized_As)-6), 'Mr. Mohammed Hanafy');
f='Success';
if w<=L3 && y1==0;
    f='Failure';
    Failures_Vector(1,w)=1;
elseif w>L3 && w<=L6 && y2==0;
    f='Failure';
    Failures_Vector(1,w)=1;
elseif w>L6 && w<=L9 && y3==0;
    f='Failure';
    Failures_Vector(1,w)=1;
end

% Tested_Im_Number=[num2str(w) '.jpg'];
% Tested_Im_Location=.........
% fullfile(Tested_Images_Folder,Tested_Im_Number);
% Tested_Im=im2double(rgb2gray(imread(Tested_Im_Location)));
% subplot(2,1,1)
% imshow(Tested_Im)
% title({'Image No.' num2str(w) ' Is Originally for';n})
% subplot(2,1,2)
% imshow(Recognized_image)
% if y1==1 || y2==1 || y3==1
%     title({'The Image Is Recognized As'.............
%         Recognized_As(1:length(Recognized_As)-6)}
%     else
%         title({'The Image Is Recognized As' ..... [an’ blanks(1) Recognized_As]}
%     end
% if y1==1 || y2==1 || y3==1
%     fprintf(fid,['%0.3d\t\t\t %s\t %s\t %s\t ',.....
Appendix A. A Code for the Digital Model

```matlab
'%s \n\n',w,n,........
Recognized_As(1:length(Recognized_As)-6),f);
Latex_Matrix(w,1:4)={num2str(w) n ........
Recognized_As(1:length(Recognized_As)-6) f);
else fprintf(fid,........
'\%0.3d\t\t\t \%22s\t \%22s\t %s \n\n',....
w,n,Recognized_As,f);
Latex_Matrix(w,1:4)={num2str(w) n Recognized_As f);
end
% disp(['Please, press Any keyboard button to see another '....
% 'face and its recognition....'])
% pause
% clc
end
Total_Number_of_Failures=sum(Failures_Vector); % The total number
% of failures in
% the face
% recognition
% process.
fprintf(fid,['==============================================='....
'======================================
']);
fprintf(fid,['** The Total Number of Successes: %0.3d out of '....
'Total_No_of_Tested_Im,Total_No_of_Failures,Total_No_of_Tested_Im,....
((Total_No_of_Tested_Im-Total_Number_of_Failures)/....
Total_No_of_Tested_Im)*100);
fprintf(fid,['** The Total Number of Failures: %0.3d out of '....
'Total_No_of_Failures,Total_No_of_Tested_Im,....
(Total_Number_of_Failures/Total_No_of_Tested_Im)*100);
fclose(fid);
close all
clc
disp(['Please, see the documented results of face recognition '....
'in the open'])
disp(['text file then press any keyboard button to resume the '....
'code >>>'])
Text='Face Recognition Results for Testing.txt';
open(Text) % Opening the text file which contains
% the results of face recognition.
pause
clc
open('PCA_IPCA_Testing_and_Setting_up_Thresholds.m')
```

%% The Computation of Face Recognition Performance 
clc
close all

Error_Rate_Recognition=........
zeros(1,length(Eigenvalues)); % Measuring recognition
% performance. The elements
% of this vector represent
% the error rates in recognition
% when different eigenfaces are
% selected.

Eigenvalues1=fliplr(Eigenvalues);
Eigenvectors1=flipdim(Eigenvectors,2); % This flipping just to make
% the highest correlated
% eigenface is associated
% with the first eigenvalue
% and so on. This is done
% to be compatible with the
% generated document
% "PCA vs. IPCA".

for kk=1:length(Eigenvalues)
    ['Iteration No.: ' num2str(kk) ' Out of ' ........
    num2str(length(Eigenvalues))]
    Selected_Eigenvectors=Eigenvectors1(:,1:kk);

    % Projecting each training face on the eigenspace. It is just
    % expressing each training face in terms of the eigenfaces.
    % This is called principal components transform (also called
    % the Hotelling or Karhunen-Loéve transform)
    All_Known_Transformed_Im=........
    zeros(size(Selected_Eigenvectors,2),........
    Total_No_of_Known_Im); % A 2D matrix where each column
    % represents the coordinates of
    % a projected training face in
    % the eigenspace.
    for r=1:Total_No_of_Known_Im
        All_Known_Transformed_Im(:,r)=Selected_Eigenvectors.'*....
        Known_Im_Subt_Mean(:,r);
    end
    All_Known_Transformed_Im;

    % Projecting each tested image on the eigenspace. It is just
    % expressing each tested image in terms of the eigenfaces. This
    % is called principal components transform (also called the
    % Hotelling or Karhunen-Loéve transform)
Appendix A. A Code for the Digital Model

All_Tested_Transformed_Im=....... zeros(size(Selected_Eigenvectors,2),...... Total_No_of_Tested_Im); % A 2D matrix where each column represents the coordinates of a projected tested image in the eigenspace.

for r1=1:Total_No_of_Tested_Im
    All_Tested_Transformed_Im(:,r1)=...............
    Selected_Eigenvectors.'*Tested_Im_Subt_Mean(:,r1);
end

All_Tested_Transformed_Im;

fid=fopen('.\Face Recognition Performance\Recognition .... 'Results for Different Eigenfaces\Results for Testing .... 'When q=' num2str(kk) '.txt','w'); % A text file for typing the results of face recognition.

fprintf(fid,['\n\t ***** The Results of Face Recognition .... ' .... 'When q=' num2str(kk) ' Obtained from the *****\r
\n\t ***** PCA and IPCA Code for Testing .... ' .... 'and Setting up Thresholds *****\r
\n\t for The Image No. The Image Is Originally .... ' .... 'The Image Is Recognized as The Status\r
\n\t =============== ========================== ==========

Distances_Vector=....... zeros(Total_No_of_Tested_Im,...... Total_No_of_Known_Im); % A 2D matrix where each row represents the distances between the weights of each training face and the weights of one of the tested images.

% The distances between the weights of a training face and the weights of each corresponding tested image. Note that, here % the training face and tested images have the same face and projection so these distances must be the smallest. The % distances are extracted from the Distances_Vector matrix.

Distances_Vector_P1=zeros(1,Im_P(1));
Distances_Vector_P2=zeros(1,Im_P(2));
Distances_Vector_P3=zeros(1,Im_P(3));
Distances_Vector_P4=zeros(1,Im_P(4));
Distances_Vector_P5=zeros(1,Im_P(5));
Distances_Vector_P6=zeros(1,Im_P(6));
Distances_Vector_P7=zeros(1,Im_P(7));
Distances_Vector_P8=zeros(1,Im_P(8));
distances_vector_p9=zeros(1,Im_P(9));
for p=1:total_no_of_tested_im
  for q=1:total_no_of_known_im
    distances_vector(p,q)=......
    norm(all_tested_transformed_im(:,p)−......
    all_known_transformed_im(:,q));
  end
  if p<=L1
    distances_vector_p1(1,p)=distances_vector(p,1);
  elseif p>L1 && p<=L2
    distances_vector_p2(1,p−L1)=distances_vector(p,2);
  elseif p>L2 && p<=L3
    distances_vector_p3(1,p−L2)=distances_vector(p,3);
  elseif p>L3 && p<=L4
    distances_vector_p4(1,p−L3)=distances_vector(p,4);
  elseif p>L4 && p<=L5
    distances_vector_p5(1,p−L4)=distances_vector(p,5);
  elseif p>L5 && p<=L6
    distances_vector_p6(1,p−L5)=distances_vector(p,6);
  elseif p>L6 && p<=L7
    distances_vector_p7(1,p−L6)=distances_vector(p,7);
  elseif p>L7 && p<=L8
    distances_vector_p8(1,p−L7)=distances_vector(p,8);
  elseif p>L8 && p<=L9
    distances_vector_p9(1,p−L8)=distances_vector(p,9);
  end
  d1=distances_vector_p1;
  d2=distances_vector_p2;
  d3=distances_vector_p3;
  d4=distances_vector_p4;
  d5=distances_vector_p5;
  d6=distances_vector_p6;
  d7=distances_vector_p7;
  d8=distances_vector_p8;
  d9=distances_vector_p9;
  % the calculation of the mean and standard deviation for each
  % vector of the minimum distances and stacking them in a vector
  % for the means and another for the standard deviations. This
  % is done for setting up a threshold for face recognition
  % because when a distance between a training face and a tested
  % image is the smallest with respect to the other training
  % faces, that does not mean the tested image is recognized as
  % the training face due to the tested image can be different
  % than the training face and has the smallest distance in the
  % same time. So, a certain threshold must be used to increase
  % the accuracy of recognition.
Appendix A. A Code for the Digital Model

\[
\text{P\_Mean=} \begin{bmatrix}
\text{mean}(d1); \text{mean}(d2); \text{mean}(d3); \text{mean}(d4); \text{mean}(d5); \\
\text{mean}(d6); \text{mean}(d7); \text{mean}(d8); \text{mean}(d9);
\end{bmatrix};
\]

\[
\text{P\_STD=} \begin{bmatrix}
\text{std}(d1); \text{std}(d2); \text{std}(d3); \text{std}(d4); \text{std}(d5); \text{std}(d6); \\
\text{std}(d7); \text{std}(d8); \text{std}(d9);
\end{bmatrix};
\]

```
Failures_Vector=zeros(1,......
    Total_No_of_Tested_Im); % A vector for counting the
    % number of failures in the
    % face recognition process.

for w=1:Total_No_of_Tested_Im
    % The face recognition process.
    for h=1:Total_No_of_Known_Im
        if min(Distances_Vector(w,:))==.....
            Distances_Vector(w,h) && .....
            min(Distances_Vector(w,:))>=.....
            (P\_Mean(h,1)−P\_STD(h,1)) && .....
            min(Distances_Vector(w,:))<=....
            (P\_Mean(h,1)+P\_STD(h,1))
            s=transpose(struct2cell(Known_Images));
            c=sortrows(s,1);
            z=c(:,1);
            Recognized_As=char(z(h,1));
            Recognized\_Image\_Location=....
            fullfile(Known\_Images\_Folder,Recognized\_As);
            Recognized\_image=im2double(rgb2gray(......
                imread(Recognized\_Image\_Location)));
            break;
        else
            Recognized\_As='Unknown Image';
            Recognized\_image=imread('Unknown\_Face.jpg');
        end
    end
    % Defining the face.
    if w<=L3
        n='Mr. Mansour Alshammari';
    elseif L3<w && w<=L6
        n='Mr. Methkir Alharthee';
    else
        n='Mr. Mohammed Hanafy';
    end

    y1=strcmp(Recognized\_As(1:length(Recognized\_As)−6),.....
        'Mr. Mansour Alshammari');
    y2=strcmp(Recognized\_As(1:length(Recognized\_As)−6),.....
```
'Mr. Methkir Alharthee');

y3=strcmp(Recognized_As(1:length(Recognized_As)-6),....
'Mr. Mohammed Hanafy');

f='Success';

if w<=L3 && y1==0;
    f='Failure';
    Failures_Vector(1,w)=1;
elseif w>L3 && w<=L6 && y2==0;
    f='Failure';
    Failures_Vector(1,w)=1;
elseif w>L6 && w<=L9 && y3==0;
    f='Failure';
    Failures_Vector(1,w)=1;
end

% clc
Tested_Im_Number=[num2str(w) '.jpg'];
Tested_Im_Location=.....
fullfile(Tested_Images_Folder,Tested_Im_Number);
Tested_Im=.....
im2double(rgb2gray(imread(Tested_Im_Location)));
subplot(2,1,1)
imshow(Tested_Im)
title({'Image No.' num2str(w) ' Is Originally for';n})
subplot(2,1,2)
imshow(Recognized_image)
if y1==1 || y2==1 || y3==1
    title({'The Image Is Recognized As';......
    Recognized_As(1:length(Recognized_As)-6);......
    ['When q=' num2str(kk)]})
else title({'The Image Is Recognized As';......
    ['an' blanks(1) Recognized_As];......
    ['When q=' num2str(kk)]})
end

if y1==1 || y2==1 || y3==1
    fprintf(fid,['%0.3d			 %s	 ' ....
    '%-22s	 %s 

' ],......
w,n,Recognized_As(1:length(Recognized_As)-6),f);
else fprintf(fid,['%0.3d			 %s	 ' ....
    '%-22s	 %s 

' ],....
w,n,Recognized_As,f);
end

disp(['Please, press Any keyboard button to see '.....
    'another face and its recognition when q=' .....
    num2str(kk) ' .......'])

pause
clc
end
Total_Number_of_Failures=
sum(Failures_Vector); % The total number of failures in
% the face recognition process.

Error_Rate_Recognition(1,kk)=(Total_Number_of_Failures/......
Total_No_of_Tested_Im)*100;

fprintf(fid,'==========================================
'
'==========================================
');

fprintf(fid,'** The Total Number of Successes: %0.3d out '....
'of %0.3d (%3.4f%%) 

',Total_No_of_Tested_Im−Total_Number_of_Failures,....
Total_No_of_Tested_Im,......
((Total_No_of_Tested_Im−Total_Number_of_Failures)/....
Total_No_of_Tested_Im)*100);

fprintf(fid,'** The Total Number of Failures: %0.3d out '....
'of %0.3d (%3.4f%%) \r\n\n',Total_Number_of_Failures,....
Total_No_of_Tested_Im,(Total_Number_of_Failures/....
Total_No_of_Tested_Im)*100);

fclose(fid);

% close all
% clc

% disp(['Please, see the documented results of face '........
% 'recognition when q=' num2str(kk) ' in the open'])
% disp(['text file then press any keyboard button to '.....
% 'resume the code >>>'])
% Text=['."Face Recognition Performance\Testing Face '.....
% 'Recognition Results for Different Eigenfaces\Face ' ....
% 'Recognition Results for Testing When q=' .....
% num2str(kk) '.txt'];
% open(Text) % Opening the text file which contains
% % the results of face recognition.
% pause
% clc
% open('PCA_IPCA_Testing_and_Setting_up_Thresholds.m')
% end

% Plotting the error rates of face recognition for different
% selected eigenfaces compared with the error rates for PCA and
% IPCA algorithms.

clc
figure('units','centimeters','position',[0.15 1.2 35.8 16.9])
subplot(2,1,1)
Leg1=plot(1:length(Eigenvalues),Error_Rate_Recognition);
hold on
Leg2=plot(1,Error_Rate_Recognition(1),'kd','LineWidth',1.5,....
'MarkerEdgeColor','k','MarkerFaceColor','m','MarkerSize',8);
text(25,Error_Rate_Recognition(1)+2,{........
  '\fontsize{10} \color{black}' Error_Rate_Recognition(1)})
hold on
Leg3=plot(length(Eigenvalues),........
  Error_Rate_Recognition(length(Eigenvalues)),'kd',.....
  'LineWidth',1.5,'MarkerEdgeColor','k','MarkerFaceColor',.....
  'g','MarkerSize',8);
text(length(Eigenvalues)−60,........
  Error_Rate_Recognition(length(Eigenvalues))+8,.....
  '{\fontsize{10} \color{black}' .......
  Error_Rate_Recognition(length(Eigenvalues))})
hold off
axis([0 length(Eigenvalues)+(length(Eigenvalues)/40) ......
  max(Error_Rate_Recognition)/10 ......
  set(gca,'XTick',[1 length(Eigenvalues)/2 length(Eigenvalues)])
title(['The Error Rates of Recognition for Different '.....
  'Selected Eigenfaces'])
xlabel('The Number of Selected Eigenfaces')
ylabel('The Error Rate (%)')
legend([Leg1 Leg2 Leg3],['The Error Rates of Recognition for '....
  'Different Selected Eigenfaces.'],
       ['The Error Rate of Recognition When the First Eigenface '....
  'Is Selected.'],['The Error Rate of Recognition When All '....
  'Eigenfaces Are Selected.'],'Location','NorthEast');
subplot(2,1,2)
Leg1=plot(1:length(Eigenvalues),Error_Rate_Recognition);
axis([0 20 0 ...........
  max(Error_Rate_Recognition)+(max(Error_Rate_Recognition)/10)])
Leg2=line([length(Best_Eigenvalues_PCA) ..... 
  length(Best_Eigenvalues_PCA)],....
  [0 Error_Rate_Recognition(length(Best_Eigenvalues_PCA))],....
  'LineStyle', '−−', 'Color', 'r', 'LineWidth', 4);
line([0 length(Best_Eigenvalues_PCA)],.....
  [Error_Rate_Recognition(length(Best_Eigenvalues_PCA))])
Error_Rate_Recognition(length(Best_Eigenvalues_PCA)) ..... 
  'LineStyle', '−−', 'Color', 'r', 'LineWidth', 4)
text(length(Best_Eigenvalues_PCA)-0.8,.....
  Error_Rate_Recognition(length(Best_Eigenvalues_PCA))+5.7,.....
  '{\fontsize{10} \color{red} \bf' ......
  num2str(Error_Rate_Recognition(length(Best_Eigenvalues_PCA)))
  Leg3=line([length(Best_Eigenvalues_IPCA) ..... 
  length(Best_Eigenvalues_IPCA)],....
  [0 Error_Rate_Recognition(length(Best_Eigenvalues_IPCA))],....
  'LineStyle', '−−', 'Color', 'k', 'LineWidth', 3);
line([0 length(Best_Eigenvalues_IPCA)],.....
  [Error_Rate_Recognition(length(Best_Eigenvalues_IPCA))])
Error_Rate_Recognition(length(Best_Eigenvalues_IPCA)) ..... 
  'LineStyle', '−−', 'Color', 'k', 'LineWidth', 3)
text(0.3,.......
Appendix A. A Code for the Digital Model

```matlab
Error_Rate_Recognition(length(Best_Eigenvalues_IPCA))+6,.....
num2str(......
Error_Rate_Recognition(length(Best_Eigenvalues_IPCA))))
vv=[length(Best_Eigenvalues_IPCA) length(Best_Eigenvalues_PCA)];
if vv(1)==1 || vv(2)==1
    set(gca,'XTick',[0 sort(vv) 20])
elser set(gca,'XTick',[0 1 sort(vv) 20])
end
title(['Explaining the Error Rates for the PCA and IPCA ' .......
'Algorithms on the Plot'])
xlabel('The Number of Selected Eigenfaces')
ylabel('The Error Rate (%)')
legend([Leg1 Leg2 Leg3],['The Error Rates of Recognition for ' .......
'Different Selected Eigenfaces.'],.....
'The Error Rate for PCA Algorithm.','....
'The Error Rate for IPCA Algorithm.','Location','NorthEast');

%%

%%% A Face Detection Process %%%%%%%%
clc
close all

%%% When we only select the eigenfaces which contain the most
%%% significant patterns from the correlated training faces then
%%% the accuracy of face detection and processing speed will
%%% increase. That because in face detection, distance
%%% calculation will be between a pre-process unknown image and
%%% its reconstruction so if a reconstructed image is not a face
%%% then it will have a big distance hence it will not be
%%% detected as a face image. As a result of that, the usage of
%%% the calculated eigenfaces by using IPCA algorithm will
%%% produce the smallest error rate and processing time then the
%%% usage of the calculated eigenfaces by using PCA algorithm
%%% comes second finally the usage of all calculated eigenvectors
%%% as eigenfaces will produce the biggest error rate and
%%% processing time. It is very important to notice that the
%%% calculation of all eigenvectors from the covariance matrix
%%% for all training faces is too difficult because the
%%% covariance matrix is too big as explained before so it is
%%% impractical to use all eigenvectors as eigenfaces.
% Best_Eigenvectors=Eigenvectors; % Just try it to see how
% affects on the results
% of face detection.
% Best_Eigenvectors=Best_Eigenvectors_PCA; % Just try it to see how
```

The code snippet above demonstrates how to calculate and visualize error rates for different eigenfaces in the context of face detection and processing. It includes a function for error rate recognition, setting XTick for the plot, and a title explaining the error rates for PCA and IPCA algorithms. The code also discusses the benefits of selecting eigenfaces that contain the most significant patterns in terms of improving detection accuracy and processing speed. It highlights the trade-offs between using different sets of eigenfaces and emphasizes the impracticality of using all eigenvectors due to computational limitations.
% affects on the results of face detection.
Best_Eigenvectors=Best_Eigenvectors_IPCA; % Just try it to see how % affects on the results of face detection.

% Projecting each training face on the eigenspace. It is just % expressing each training face in terms of the eigenfaces. This is % called principal components transform (also called the Hotelling % or Karhunen-Loève transform)
All_Known_Transformed_Im=.......
    zeros(size(Best_Eigenvectors,2),......
    Total_No_of_Known_Im); % A 2D matrix where each column % represents the coordinates of % a projected training face in % the eigenspace.
for r=1:Total_No_of_Known_Im
    All_Known_Transformed_Im(:,r)=Best_Eigenvectors.'*.......    
        Known_Im_Subt_Mean(:,r);
end
All_Known_Transformed_Im;

% Projecting each tested image on the eigenspace. It is just % expressing each tested image in terms of the eigenfaces. This is % called principal components transform (also called the Hotelling % or Karhunen-Loève transform)
All_Tested_Transformed_Im=........
    zeros(size(Best_Eigenvectors,2),......
    Total_No_of_Tested_Im); % A 2D matrix where each column % represents the coordinates of % a projected tested image in % the eigenspace.
for r1=1:Total_No_of_Tested_Im
    All_Tested_Transformed_Im(:,r1)=Best_Eigenvectors.'*.......    
        Tested_Im_Subt_Mean(:,r1);
end
All_Tested_Transformed_Im;

fid=fopen('Face Detection Results for Testing.txt',........
    'w'); % A text file for typing the % results of face detection.
fprintf(fid,['\n\t ***** The Results of Face Detection '........
    'Obtained from the PCA and IPCA *****\n']);
fprintf(fid,['\n\t			 ***** Code for Testing and Setting '........
    'up Thresholds *****\n']);
fprintf(fid,['\n\tImage No.\t The Image Originally Is\t The Detected Image Is\t The Status\n']);
fprintf(fid,["=============	 ==============\t \\
'=============	 =========\n']);

% Note that, the tested images must be unknown whether they are
% face images or not but they are known here to be face images just
% for testing the face detection process and for setting up a
% detection threshold.
Distances_Vector1=zeros(1,......
    Total_No_of_Tested_Im); % A vector contains the distances
    % between each pre-processed tested
    % image and its reconstruction.
for i=1:Total_No_of_Tested_Im
    g=Best_Eigenvectors*All_Tested_Transformed_Im(:,i);
    Reconstructed_Tested_Im=max(Tested_Im_Subt_Mean(:,i))*.......
    (double(g)/max(max(g))); % This is for making each
    % pre-processed tested image
    % and its reconstruction have
    % approximately the same
    % dynamic range.
    Distances_Vector1(1,i)=norm(Tested_Im_Subt_Mean(:,i)....
        Reconstructed_Tested_Im); % Note that, a calculated
        % distance must be between
        % a pre-processed tested
        % image and its reconstruction
        % but it is not between an
        % original tested image and
        % its reconstruction.
end

% The calculation of the mean and standard deviation for the
% calculated distances between each pre-processed tested image and
% its reconstruction. This is done for setting up a threshold for
% face detection.
Mean=mean(Distances_Vector1);
Std=std(Distances_Vector1);
% save(["Computed Mean for Face Detection When All ",
    'Eigenvectors Are Used as Eigenfaces','
",'Mean'])
% save(["Computed STD for Face Detection When All 
",
    'Eigenvectors Are Used as Eigenfaces','
",'Std'])
% save(["Computed Mean for Face Detection When the Eigenfaces ",
    'Computed by Using PCA Algorithm Are Used','
",'Mean'])
% save(["Computed STD for Face Detection When the Eigenfaces 
",'Computed by Using PCA Algorithm Are Used','
",'Std'])
% save(["Computed Mean for Face Detection When the Eigenfaces ",
    'Computed by Using IPCA Algorithm Are Used','
",'Mean'])
% save(["Computed STD for Face Detection When the Eigenfaces 
",'Computed by Using IPCA Algorithm Are Used','
",'Std'])
Failures_Vector1=zeros(1,......
Total_No_of_Tested_Im;  % A vector for counting the
    % number of failures in the
    % face detection process.

Latex_Matrix=cell(Total_No_of_Tested_Im,4);  % This matrix is used
    % for creating a table
    % in Latex.

for w1=1:Total_No_of_Tested_Im
    % The face detection process.
    if Distances_Vector1(1,w1)>=(Mean−Std) && ......
        Distances_Vector1(1,w1)<=(Mean+Std)
        Detected_As='a face';
        Detected_Image=imread('A_Face.jpg');
    else
        Detected_As='not a face';
        Detected_Image=imread('Not_a_Face.jpg');
    end

    b='a face';  % Defining an original tested image.
    f1='Success';
    e=strcmp(Detected_As,'a face');
    if w1<=L9 && e==0
        f1='Failure';
        Failures_Vector1(1,w1)=1;
    end

    % Tested_Im_Number1=[num2str(w1) '.jpg'];
    % Tested_Im_Location1=fullfile(Tested_Images_Folder,.....
    % Tested_Im1=im2double(rgb2gray(imread(Tested_Im_Location1)));
    % subplot(2,1,1)
    % imshow(Tested_Im1)
    % title(['Image No.' num2str(w1) ' Originally Is'])
    % subplot(2,1,2)
    % imshow(Detected_Image)
    % title('It Is Detected As')
    % disp(['Please, press any keyboard button to see another '....
    % 'image and its detection....'])
    % pause

    if e==1
        fprintf(fid,['%0.3d\t\t\t\t %s\t\t\t\t %s\t\t\t\t %s\t\t\t\t\t %s
\n
',w1,b,Detected_As,f1);
        Latex_Matrix(w1,1:4)={num2str(w1) b Detected_As f1};
    else fprintf(fid,['%0.3d\t\t\t\t %s\t\t\t\t %s\t\t\t\t %s\t\t\t\t\t %s
\n
',w1,b,Detected_As,f1);
        Latex_Matrix(w1,1:4)={num2str(w1) b Detected_As f1};
    end
Appendix A. A Code for the Digital Model

```matlab
% clc
end
Total_Number_of_Failures1=......
    sum(Failures_Vector1); % The total number of failures
    % in the face detection process.
fprintf(fid,['==============================================='....
    '===================================
'],Total_No_of_Tested_Im,....
    Total_Number_of_Failures1,Total_No_of_Tested_Im,....
    ((Total_No_of_Tested_Im-Total_Number_of_Failures1)/....
    Total_No_of_Tested_Im)*100);
fprintf(fid,['** The Total Number of Failures: %0.3d out '......
    'of %0.3d (%3.4f%%) 

'],Total_Number_of_Failures1,......
    Total_No_of_Tested_Im,........
    (Total_Number_of_Failures1/Total_No_of_Tested_Im)*100);
fclose(fid);
close all
clc
disp(['Please, see the documented results of face detection '....
    'in the open'])
disp(['text file then press any keyboard button to resume '......
    'the code >>>'])
Text='Face Detection Results for Testing.txt';
open(Text) % Opening the text file which contains
    % the results of face detection.
pause
clc
open('PCA_IPCA_Testing_and_Setting_up_Thresholds.m')

%% The Computation of Face Detection Performance %%%%%%%
clc
close all
close
Error_Rate_Detection=.......% Measuring detection
    % performance. The elements
    % of this vector represent
    % the error rates in detection
    % when different eigenfaces are
    % selected.
Eigenvalues1=flipr(Eigenvalues);
```
Eigenvectors1 = flipdim(Eigenvectors,2); % This flipping just to make
% the highest correlated
% eigenface is associated
% with the first eigenvalue
% and so on. This is done to
% be compatible with the
% generated document
% "PCA vs. IPCA".

for kk=1:length(Eigenvalues)
    ['Iteration No.: ' num2str(kk) ' Out of ' ..... num2str(length(Eigenvalues))]
    Selected_Eigenvectors=Eigenvectors1(:,1:kk);

    % Projecting each training face on the eigenspace. It is just
    % expressing each training face in terms of the eigenfaces.
    % This is called principal components transform (also called
    % the Hotelling or Karhunen-Loéve transform)
    All_Known_Transformed_Im=........
    zeros(size(Selected_Eigenvectors,2),........
    Total_No_of_Known_Im); % A 2D matrix where each column
    % represents the coordinates of a
    % projected training face in the
    % eigenspace.
    for r=1:Total_No_of_Known_Im
        All_Known_Transformed_Im(:,r)=Selected_Eigenvectors.'*....
        Known_Im_Subt_Mean(:,r);
    end
    All_Known_Transformed_Im;

    % Projecting each tested image on the eigenspace. It is just
    % expressing each tested image in terms of the eigenfaces. This
    % is called principal components transform (also called the
    % Hotelling or Karhunen-Loéve transform)
    All_Tested_Transformed_Im=........
    zeros(size(Selected_Eigenvectors,2),........
    Total_No_of_Tested_Im); % A 2D matrix where each column
    % represents the coordinates of a
    % projected tested image in the
    % eigenspace.
    for r1=1:Total_No_of_Tested_Im
        All_Tested_Transformed_Im(:,r1)=.....
        Selected_Eigenvectors.'*Tested_Im_Subt_Mean(:,r1);
    end
    All_Tested_Transformed_Im;

    fid=fopen(['\Face Detection Performance\Detection '..... 'Results for Different Eigenfaces\Results for Testing '....

}
% A text file for typing the results of face Detection.
fprintf(fid, ['
***** The Results of Face Detection ' ....
'When q=' num2str(kk) 'txt'], 'w');
% Obtained from the PCA and *****
fprintf(fid, ['\n***** IPCA Code for Testing and ' ....
'Setting up Thresholds *****'], 'w');

% Is \t The Detected Image Is | The Status |

fprintf(fid, ['Image No.\t The Image Originally ' ....
'Total_No_of_Tested_Im); % A vector contains the distances
% between each pre-processed tested image and its reconstruction.

for i=1:Total_No_of_Tested_Im
    g=Selected_Eigenvectors*All_Tested_Transformed_Im(:,i);
    Reconstructed_Tested_Im=max(Tested_Im_Subt_Mean(:,i))*....
        (double(g)/max(max(g))); % This is for making each pre-processed tested image and its reconstruction have approximately the same dynamic range.
    Distances_Vector1(i,i)=norm(Tested_Im_Subt_Mean(:,i) - Reconstructed_Tested_Im); % Note that, a calculated distance must be between a pre-processed tested image and its reconstruction but it is not between an original tested image and its reconstruction.
end

% The calculation of the mean and standard deviation for the calculated distances between each pre-processed tested image and its reconstruction. This is done for setting up a threshold for face detection.
Mean=mean(Distances_Vector1);
Std=std(Distances_Vector1);

save([cd '\Face Detection Performance\Means and STDs for ' ....
'Different Eigenfaces\Mean for Face Detection When q='], num2str(kk)), 'Mean')

save([cd '\Face Detection Performance\Means and STDs for ' ....
Different Eigenfaces\STD for Face Detection When q=' ...... num2str(kk)}, 'Std')

Failures_Vector1=zeros(1,......
Total_No_of_Tested_Im); % A vector for counting the % number of failures in the % face detection process.

for w1=1:Total_No_of_Tested_Im
  % The face detection process.
  if Distances_Vector1(1,w1)>=(Mean−Std) && ......
    Distances_Vector1(1,w1)<=(Mean+Std)
    Detected_As='a face';
    Detected_Image=imread('A_Face.jpg');
  else
    Detected_As='not a face';
    Detected_Image=imread('Not_a_Face.jpg');
  end

b='a face'; % Defining an original tested image.

f1='Success';
e=strcmp(Detected_As,'a face');
if w1<=L9 && e==0
  f1='Failure';
  Failures_Vector1(1,w1)=1;
end

clc
Tested_Im_Number1=[num2str(w1) '.jpg'];
Tested_Im_Location1=fullfile(Tested_Images_Folder,....
Tested_Im_Number1);
Tested_Im1=im2double(......
  rgb2gray(imread(Tested_Im_Location1)));
subplot(2,1,1)
imshow(Tested_Im1)
title(['Image No.' num2str(w1) ' Originally Is'])
subplot(2,1,2)
imshow(Detected_Image)
title(['It Is Detected As';['(When q=' num2str(kk) ')']])

if e==1
  fprintf(fid,['%0.3d	%3s	%3s 
',w1,b,Detected_As,f1);
else fprintf(fid,['%0.3d	%3s	%3s 
',w1,b,Detected_As,f1);
end

disp(['Please, press any keyboard button to see ...... ' .......
  'another image and its detection when q=' ......
  num2str(kk) ' .......'])
Appendix A. A Code for the Digital Model

```matlab
% pause
% clc
end
Total_Number_of_Failures1=....... sum(Failures_Vector1); % The total number of failures in the face detection process.

Error_Rate_Detection(1,kk)=....... (Total_Number_of_Failures1/Total_No_of_Tested_Im)*100;

fprintf(fid,['========================================='..... '========================================
']);
fprintf(fid,['** The Total Number of Successes: %0.3d '..... 'out of %0.3d (%3.4f%%) 

'],...
Total_No_of_Tested_Im−Total_Number_of_Failures1,.... Total_No_of_Tested_Im,....... ((Total_No_of_Tested_Im−Total_Number_of_Failures1)/...... Total_No_of_Tested_Im)*100);

fprintf(fid,['** The Total Number of Failures: %0.3d '..... 'out of %0.3d (%3.4f%%) 

'],.... Total_Number_of_Failures1,Total_No_of_Tested_Im,....... (Total_Number_of_Failures1/Total_No_of_Tested_Im)*100);
fclose(fid);

% close all
% clc
% disp(['Please, see the documented results of face '..... 'detection when q=' num2str(kk) ' in the open'])
% disp(['text file then press any keyboard button to '..... 'resume the code >>>'])
% Text=['.\Face Detection Performance\Testing Face '..... 'Detection Results for Different Eigenfaces\Face '...... 'Detection Results for Testing When q=' ......
num2str(kk) '.txt'];
% open(Text) % Opening the text file which contains % the results of face detection.
% pause
% clc
% open('PCA_IPCA_Testing_and_Setting_up_Thresholds.m')
end

% Plotting the error rates of face detection for different selected % eigenfaces compared with the error rates for the PCA and IPCA % algorithms.
clc
figure('units','centimeters','position',[0.15 1.2 35.8 16.9])
subplot(2,1,1)
Leg1=plot(1:length(Eigenvalues),Error_Rate_Detection);
```
hold on
Leg2=plot(1,Error_Rate_Detection(1),'kd','LineWidth',1.5,...
'MarkerEdgeColor','k','MarkerFaceColor','m','MarkerSize',8);
text(24.8,Error_Rate_Detection(1)+1.8,...
{'\fontsize{10} \color{black}' Error_Rate_Detection(1)})
hold on
Leg3=plot(length(Eigenvalues),...
Error_Rate_Detection(length(Eigenvalues)),'kd',....
'LineWidth',1.5,'MarkerEdgeColor','k',....
'MarkerFaceColor','g','MarkerSize',8);
text(length(Eigenvalues)+55,....
Error_Rate_Detection(length(Eigenvalues))-1.8,....
{'\fontsize{10} \color{black}' Error_Rate_Detection(length(Eigenvalues))})
hold off
axis([0 length(Eigenvalues)+(length(Eigenvalues)/40) ....
-max(length(Eigenvalues))/10 ...........
max(length(Eigenvalues)+(max(length(Eigenvalues))/10))])
set(gca,'XTick',[1 length(Eigenvalues)/2 length(Eigenvalues)])
title(['The Error Rates of Detection for Different '....
'Selected Eigenfaces'])
xlabel('The Number of Selected Eigenfaces')
ylabel('The Error Rate (%)')
legend([Leg1 Leg2 Leg3],['The Error Rates of Detection for '.....
'Different Selected Eigenfaces.'],['The Error Rate of '....
'Detection When the First Eigenface Is Selected.'],....
['The Error Rate of Detection When All Eigenfaces Are '.....
'Selected.'],'Location','NorthWest');

Leg1=plot(1:length(Eigenvalues),Error_Rate_Detection);
axis([0 20 0 .......]
max(length(Eigenvalues)+(max(length(Eigenvalues))/10)))]
Leg2=line([length(Best_Eigenvalues_PCA) ....
length(Best_Eigenvalues_PCA)],....
[0 Error_Rate_Detection(length(Best_Eigenvalues_PCA))],....
'LineStyle','--','Color','k','LineWidth',3);
line([0 length(Best_Eigenvalues_PCA)],....;
[Error_Rate_Detection(length(Best_Eigenvalues_PCA)) ....
Error_Rate_Detection(length(Best_Eigenvalues_PCA))],....
'LineStyle','--','Color','k','LineWidth',3)
text(length(Best_Eigenvalues_PCA)-0.9,.......
Error_Rate_Detection(length(Best_Eigenvalues_PCA))+3.6,....
{'\fontsize{10} \color{black}' .......
num2str(Error_Rate_Detection(length(Best_Eigenvalues_PCA)))]
Leg3=line([length(Best_Eigenvalues_IPCA) ....
length(Best_Eigenvalues_IPCA)],....
[0 Error_Rate_Detection(length(Best_Eigenvalues_IPCA))],....
'LineStyle','--','Color','r','LineWidth',4);
line([0 length(Best_Eigenvalues_IPCA)],....;
[Error_Rate_Detection(length(Best_Eigenvalues_IPCA)) ....
Appendix A. A Code for the Digital Model

```matlab
Error_Rate_Detection(length(Best_Eigenvalues_IPCA))), ....
LineStyle', '--', 'Color', 'r', 'LineWidth', 4)
text(0.3, ....
Error_Rate_Detection(length(Best_Eigenvalues_IPCA)) + 3.6, ....
{'\fontsize{10} \color{red} \bf' ....
num2str(Error_Rate_Detection(length(Best_Eigenvalues_IPCA)))))
vv = [length(Best_Eigenvalues_IPCA) length(Best_Eigenvalues_PCA)];
if vv(1) == 1 || vv(2) == 1
    set(gca, 'XTick', [0 sort(vv) 20])
else
    set(gca, 'XTick', [0 1 sort(vv) 20])
end
title(['Explaining the Error Rates for the PCA and IPCA ' ......
'Algorithms on the Plot'])
xlabel('The Number of Selected Eigenfaces')
ylabel('The Error Rate (%)')
legend([Leg1 Leg2 Leg3], ['The Error Rates of Detection for ' ......
'Different Selected Eigenfaces.'], ....
'The Error Rate for PCA Algorithm.', ....
'The Error Rate for IPCA Algorithm.', 'Location', 'NorthEast');
```
Appendix B

Databases for the Digital and Optical Models

All images in Figure B.1, Figure B.2, Figure B.3, Figure B.4, Figure B.5, Figure B.6, Figure B.7, Figure B.8, and Figure B.9 form the database of the tested images. The database of the objects consists of images that have vertical faces to people’s shoulders shown in Figure B.1, Figure B.4, and Figure B.7.

Figure B.1: Images for Mr. Mansour Alshammari, 1st projection.
Figure B.2: Images for Mr. Mansour Alshammari, 2\textsuperscript{nd} projection.

Figure B.3: Images for Mr. Mansour Alshammari, 3\textsuperscript{rd} projection.

Figure B.4: Images for Mr. Methkir Alharthee, 1\textsuperscript{st} projection.
Figure B.5: Images for Mr. Methkir Alharthee, 2\textsuperscript{nd} projection.

Figure B.6: Images for Mr. Methkir Alharthee, 3\textsuperscript{rd} projection.

Figure B.7: Images for Mr. Mohammed Hanafy, 1\textsuperscript{st} projection.
Appendix B. Databases for the Digital and Optical Models

Figure B.8: Images for Mr. Mohammed Hanafy, 2\textsuperscript{nd} projection.

Figure B.9: Images for Mr. Mohammed Hanafy, 3\textsuperscript{rd} projection.
Table C.1: The recognition of all 180 tested images by using the PCA algorithm.

<table>
<thead>
<tr>
<th>Tested Image No.</th>
<th>Input Face</th>
<th>Recognized Output Face</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>2</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>3</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>4</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>5</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>6</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>7</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>8</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>9</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>10</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>11</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>12</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>13</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>14</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>15</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>16</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>17</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>18</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>19</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>20</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>21</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
</tbody>
</table>

Continued on the next page...
### Appendix C. Results of the Digital Recognition

<table>
<thead>
<tr>
<th>Tested Image No.</th>
<th>Input Face</th>
<th>Recognized Output Face</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>23</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>24</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>25</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>26</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>27</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>28</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>29</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>30</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>31</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>32</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>33</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>34</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>35</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>36</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>37</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>38</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>39</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>40</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>41</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>42</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>43</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>44</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>45</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>46</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>47</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>48</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>49</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>50</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>51</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>52</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>53</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>54</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>55</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>56</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>57</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>58</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>59</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
</tbody>
</table>

Continued on the next page ...
<table>
<thead>
<tr>
<th>Tested Image No.</th>
<th>Input Face</th>
<th>Recognized Output Face</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mansour Alshammari</td>
<td>Success</td>
</tr>
<tr>
<td>61</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>62</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>63</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>64</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>65</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>66</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>67</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>68</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>69</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>70</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>71</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>72</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>73</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>74</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>75</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>76</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>77</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>78</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>79</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>80</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>81</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>82</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>83</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>84</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>85</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>86</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>87</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>88</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>89</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>90</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>91</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>92</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>93</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>94</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>95</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>96</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>97</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
</tbody>
</table>

Continued on the next page ...
### Appendix C. Results of the Digital Recognition Test

<table>
<thead>
<tr>
<th>Tested Image No.</th>
<th>Input Face</th>
<th>Recognized Output Face</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>98</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>99</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>100</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>101</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>102</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>103</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>104</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>105</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>106</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>107</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>108</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>109</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>110</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>111</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>112</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>113</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>114</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>115</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>116</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>117</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>118</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>119</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>120</td>
<td>Mr. Methkir Alharthee</td>
<td>Mr. Methkir Alharthee</td>
<td>Success</td>
</tr>
<tr>
<td>121</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>122</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>123</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>124</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>125</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>126</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>127</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>128</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>129</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>130</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>131</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>132</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>133</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>134</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>135</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
</tbody>
</table>

*Continued on the next page...*
<table>
<thead>
<tr>
<th>Tested Image No.</th>
<th>Input Face</th>
<th>Recognized Output Face</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>136</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>137</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>138</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>139</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>140</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>141</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>142</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>143</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>144</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>145</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>146</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>147</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>148</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>149</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>150</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>151</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>152</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>153</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>154</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>155</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>156</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>157</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>158</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>159</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>160</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>161</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>162</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>163</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mansour Alshammari</td>
<td>Failure</td>
</tr>
<tr>
<td>164</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>165</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>166</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>167</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>168</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>169</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Image</td>
<td>Failure</td>
</tr>
<tr>
<td>170</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>171</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>172</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>173</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
</tbody>
</table>

Continued on the next page ...
## Appendix C. Results of the Digital Recognition

<table>
<thead>
<tr>
<th>Tested Image No.</th>
<th>Input Face</th>
<th>Recognized Output Face</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>174</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>175</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>176</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>177</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>178</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>179</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>180</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
</tbody>
</table>

The table end.
## Appendix D

### Results of the Digital Detection

**Table D.1:** The detection of all 180 tested images by using the PCA algorithm.

<table>
<thead>
<tr>
<th>Tested Image No.</th>
<th>Input Image</th>
<th>Detected Output Image</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>2</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>3</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>4</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>5</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>6</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>7</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>8</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>9</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>10</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>11</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>12</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>13</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>14</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>15</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>16</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>17</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>18</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>19</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>20</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>21</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
</tbody>
</table>

Continued on the next page ...
## Appendix D. Results of the Digital Detection

<table>
<thead>
<tr>
<th>Tested Image No.</th>
<th>Input Image</th>
<th>Detected Output Image</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>23</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>24</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>25</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>26</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>27</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>28</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>29</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>30</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>31</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>32</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>33</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>34</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>35</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>36</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>37</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>38</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>39</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>40</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>41</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>42</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>43</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>44</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>45</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>46</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>47</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>48</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>49</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>50</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>51</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>52</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>53</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>54</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>55</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>56</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>57</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>58</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>59</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
</tbody>
</table>

Continued on the next page ...
<table>
<thead>
<tr>
<th>Tested Image No.</th>
<th>Input Image</th>
<th>Detected Output Image</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>61</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>62</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>63</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>64</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>65</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>66</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>67</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>68</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>69</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>70</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>71</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>72</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>73</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>74</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>75</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>76</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>77</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>78</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>79</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>80</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>81</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>82</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>83</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>84</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>85</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>86</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>87</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>88</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>89</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>90</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>91</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>92</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>93</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>94</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>95</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>96</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>97</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
</tbody>
</table>

Continued on the next page ...
### Appendix D. Results of the Digital Detection

<table>
<thead>
<tr>
<th>Tested Image No.</th>
<th>Input Image</th>
<th>Detected Output Image</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>98</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>99</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>100</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>101</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>102</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>103</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>104</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>105</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>106</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>107</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>108</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>109</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>110</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>111</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>112</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>113</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>114</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>115</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>116</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>117</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>118</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>119</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>120</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>121</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>122</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>123</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>124</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>125</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>126</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>127</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>128</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>129</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>130</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>131</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>132</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>133</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>134</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>135</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
</tbody>
</table>

Continued on the next page ...
<table>
<thead>
<tr>
<th>Tested Image No.</th>
<th>Input Image</th>
<th>Detected Output Image</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>136</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>137</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>138</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>139</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>140</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>141</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>142</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>143</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>144</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>145</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>146</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>147</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>148</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>149</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>150</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>151</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>152</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>153</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>154</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>155</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>156</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>157</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>158</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>159</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>160</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>161</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>162</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>163</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>164</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>165</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>166</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>167</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>168</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>169</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>170</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>171</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>172</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>173</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
</tbody>
</table>

Continued on the next page ...
Appendix D. Results of the Digital Detection

<table>
<thead>
<tr>
<th>Tested Image No.</th>
<th>Input Image</th>
<th>Detected Output Image</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>174</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>175</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>176</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>177</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>178</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>179</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>180</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
</tbody>
</table>

The table end.
Appendix E

Derivation of $U_2(x_2, y_2)$ for the JTC

In the joint transform correlator (JTC), the derivation of the complex amplitude distribution $U_2(x_2, y_2)$ of the Fourier transformed field in the back focal plane $P_2$ can be found as follows,

$$U_2(x_2, y_2) = \frac{1}{j\lambda f} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U_1(x_1, y_1) \exp^{-j \frac{2\pi}{\lambda f} (x_1 x_2 + y_1 y_2)} \, dx_1 dy_1$$

$$= A \frac{1}{j\lambda f} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x_1, y_1 - \frac{Y}{2}) \exp^{-j \frac{2\pi}{\lambda f} (x_1 x_2 + y_1 y_2)} \, dx_1 dy_1 +$$

$$+ A \frac{1}{j\lambda f} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x_1, y_1 + \frac{Y}{2}) \exp^{-j \frac{2\pi}{\lambda f} (x_1 x_2 + y_1 y_2)} \, dx_1 dy_1$$

Changing variables: $y_1 - \frac{Y}{2} \rightarrow a, \, dy_1 \rightarrow da \&$

$y_1 + \frac{Y}{2} \rightarrow b, \, dy_1 \rightarrow db$
Appendix E. Derivation of $U_2(x_2, y_2)$ for the JTC

\[ U_2(x_2, y_2) = \frac{A}{j\lambda f} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x_1, a) \exp^{-j\frac{2\pi}{\lambda f}(x_1x_2 + (a + y/2)y_2)} \, dx_1 \, da + \]
\[ + \frac{A}{j\lambda f} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x_1, b) \exp^{-j\frac{2\pi}{\lambda f}(x_1x_2 + (b - y/2)y_2)} \, dx_1 \, db \]

\[ = \frac{A}{j\lambda f} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x_1, a) \exp^{-j\frac{2\pi}{\lambda f}(x_1x_2 + ay_2) - j\frac{2\pi}{\lambda f} y_2} \, dx_1 \, da + \]
\[ + \frac{A}{j\lambda f} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x_1, b) \exp^{-j\frac{2\pi}{\lambda f}(x_1x_2 + by_2) + j\frac{2\pi}{\lambda f} y_2} \, dx_1 \, db \]

\[ = \frac{A}{j\lambda f} \exp^{-j\frac{\pi}{\lambda f} y_2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x_1, a) \exp^{-j\frac{2\pi}{\lambda f}(x_1x_2 + ay_2)} \, dx_1 \, da + \]
\[ + \frac{A}{j\lambda f} \exp^{j\frac{\pi}{\lambda f} y_2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x_1, b) \exp^{-j\frac{2\pi}{\lambda f}(x_1x_2 + by_2)} \, dx_1 \, db \]

Therefore,

\[ U_2(x_2, y_2) = \frac{A}{j\lambda f} \exp^{-j\frac{\pi}{\lambda f} y_2} H \left( \frac{x_2}{\lambda f}, \frac{y_2}{\lambda f} \right) + \frac{A}{j\lambda f} \exp^{j\frac{\pi}{\lambda f} y_2} G \left( \frac{x_2}{\lambda f}, \frac{y_2}{\lambda f} \right) \]
This appendix presents the joint transform correlator (JTC) code for object detection and face recognition.

```matlab
clc
clear all
close all

Impulses_Folder=['cd ...... '
'The Black Background Impulses']; % The folder of the black background impulses.
% Impulses_Folder=['cd ...... '
% 'The White Background Impulses']; % The folder of the white background impulses.
if isdir(Impulses_Folder)==0
    Error_Message=sprintf(['Error: The following folder does '.... '
    'not exist\n$\$'],Impulses_Folder);
    warndlg(Error_Message);
    break
end
Impulses=dir(fullfile(Impulses_Folder,'*.jpg')); % Listing the folder of the black or white
```
% background impulses.

% The total number of the impulses.

% Imhist for setting up a threshold to work on just the pixels of a face and throwing the background pixels. Imhist calculates the number of pixels in an impulse that have the same intensity levels. So, if an impulse has a unified background then the biggest histogram of the intensity levels will be for the background pixels because the total number of pixels that have the same intensity levels are the background pixels of the impulse. Note that, the histogram of a digital image is defined as the discrete function, \( h(r_k) = n_k \), where \( r_k \) is the kth intensity level and \( n_k \) is the number of pixels in the image whose intensity level is \( r_k \).

\[\text{hist} \_\text{Impulses} = \begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} (\text{Total} \_\text{No} \_\text{of} \_\text{Impulses}, 256);\]

\[\text{for} \; A=1: \text{Total} \_\text{No} \_\text{of} \_\text{Impulses}\]

\[\text{One} \_\text{Impulse} = \text{Impulses}(A).\text{name};\]

\[\text{Impulse} = \text{fullfile} (\text{Impulses} \_\text{Folder}, \text{One} \_\text{Impulse});\]

\[\text{Impulse} = \text{double} (\text{rgb2gray} (\text{...}) \text{imread} (\text{Impulse} \_\text{Location})); \; \text{The impulse response.}\]

\[\text{hist} \_\text{Impulses}(A,:) = \begin{bmatrix} \text{imhist} (\text{uint}8(\text{Impulse})) \end{bmatrix}; \; \text{Note that, each impulse must be scaled between 0 to 255 before using imhist. For doing that, uint8 can be used for converting the impulse class form double to uint8.}\]

\[\text{plot} (\text{hist} \_\text{Impulses}(A,:))\]

\[\text{if} \; A=1\]

\[\text{title} ([\text{\'The Histogram of The First Impulse for } \text{...} \text{Mr. Mansour Alshammari\'}])\]

\[\text{elseif} \; A=2\]

\[\text{title} ([\text{\'The Histogram of The Second Impulse for } \text{...} \text{Mr. Methkir Alharthee\'}])\]

\[\text{elseif} \; A=3\]

\[\text{title} ([\text{\'The Histogram of The Third Impulse for } \text{...} \text{Mr. Mohammed Hanafy\'}])\]

\[\text{end}\]

\[\text{xlabel} (\text{\'Intensity Level } r_{(k)}\text{\')}\]

\[\text{ylabel} ([\text{\'The Number of Pixels in the Impulse Whose } \text{...} \text{Intensity Level Is } r_{(k)} \text{ Where } h(r_{(k)}) = n_{(k)}\'])\]

\[\text{axis tight}\]

\[\text{disp} ([\text{\'Please, press any keyboard button to explore } \text{...}])\]
Mean_hist_Impulses=sum(hist_Impulses,1)/Total_No_of_Impulses; % The average histogram for all impulses.
plot(Mean_hist_Impulses)
title('The Mean Histogram of All Impulses')
xlabel('Intensity Level r_{k}'); ylabel('The Mean Number of Pixels from All Impulses Whose Intensity Level Is r_{k}');
axis tight
pause

% Setting up a threshold in order to work on just the pixels of the faces and blocking the pixels of the backgrounds.
Impulses_Threshold=8; % The picked threshold is based on the average histogram for all impulses when the impulses have black backgrounds. Note that, all intensity levels below the threshold represent the impulses backgrounds because these levels have the biggest histogram.
Impulses_Threshold=180; % The Picked threshold is based on the average histogram for all impulses when the impulses have white backgrounds. % Note that, all intensity levels above the threshold represent the impulses backgrounds because these levels have the biggest histogram.

Objects_Folder=[cd ...
'The Black Background Tested Objects']; % The folder of the black background tested objects.
				% Objects_Folder=[cd ........
				'The White Background Tested Objects']; % The folder of the white background tested objects.
				if isdir(Objects_Folder)==0
					Error_Message=sprintf(['Error: The following folder does not exist
%s'],Objects_Folder);
					warndlg(Error_Message);
					break
				end
																
Objects=dir(fullfile(Objects_Folder,'*.jpg')); % Listing the folder of the black or
Appendix F. A Code for the Optical Model

% white background
% objects.
Total_No_of_Objects=length(Objects); % The total number
% of the objects.
Objs=[36  36  36]; % Each element in this vector represents
% the total number of the objects that
% are taken for each impulse.
L1=Objs(1); % L1=60 is the total number of the
% objects for Mr. Mansour Alshammari.
L2=L1+Objs(2); % L2=120 is the total number of the
% objects for Mr. Methkir Alharthee.
L3=L2+Objs(3); % L3=180 is the total number of the
% objects for Mr. Mohammed Hanafy.

% Imhist for setting up a threshold to work on just the pixels of a
% face and throwing the background pixels. Imhist calculates the
% number of pixels in an object that have the same intensity
% levels. So, if an object has a unified background then the
% biggest histogram of the intensity levels will be for the
% background pixels because the total number of pixels that have
% the same intensity levels are the background pixels of the
% object. Note that, the histogram of a digital image is defined as
% the discrete function, h(rk)=nk, where rk is the kth intensity
% level and nk is the number of pixels in the image whose intensity
% level is rk.
hist_Objects=zeros(Total_No_of_Objects,256);
for A1=1:Total_No_of_Objects

Object_Number=[num2str(A1) '.jpg'];
Object_Location=fullfile(Objects_Folder,Object_Number);
Object=double(rgb2gray(imread(Object_Location))); % The object.

hist_Objects(A1,:)=imhist(uint8(Object)); % Note that, each
% object must be
% scaled between 0 to
% 255 before using
% imhist. For doing
% that, uint8 can be
% used for converting
% the object class
% form double to
% uint8.

plot(hist_Objects(A1,:))
if A1<=L1
    title(['The Histogram of Object No.' num2str(A1) ....
% ' for Mr. Mansour Alshammari'])
end
% elseif A1>L1 && A1<=L2
  title(['The Histogram of Object No.' num2str(A1) ' for Mr. Methkir Alharthee'])
% elseif A1>L2 && A1<=L3
  title(['The Histogram of Object No.' num2str(A1) ' for Mr. Mohammed Hanafy'])
end
xlabel('Intensity Level r_{k}');
ylabel({'The Number of Pixels in the Object Whose ' ....
  'Intensity Level Is r_{k} Where h(r_{k})=n_{k}'})
axis tight

disp(['Please, press any keyboard button to explore ' ......
  'the remaining histograms >>>>>>>'])
pause
clc
end

Mean_hist_Objects=sum(hist_Objects, 1)/............
  Total_No_of_Objects; % The average histogram
  % for all objects.
plot(Mean_hist_Objects)
title('The Mean Histogram of All Objects')
xlabel('Intensity Level r_{k}');
ylabel({'The Mean Number of Pixels from All Objects Whose'; ....
  'Intensity Level Is r_{k}'})
axis tight
pause

% Setting up a threshold in order to work on just the pixels of the
% faces and blocking the pixels of the backgrounds.
Objects_Threshold=8; % The picked threshold is based on the average
  % histogram for all objects when the objects
  % have black backgrounds. Note that, all
  % intesity levels below the threshold
  % represent the objects backgrounds because
  % these levels have the biggest histogram.
Objects_Threshold=180; % The Picked threshold is based on the
  % average histogram for all objects when
  % the objects have white backgrounds. Note
  % that, all intesity levels above the
  % threshold represent the objects
  % backgrounds because these levels have
  % the biggest histogram.

Max_Desired_Cross_Fields=zeros(Total_No_of_Objects, ....
  Total_No_of_Impulses); % Each element of each row in this
  % matrix represents the maximum value
  % of the desired crosscorrelated field
for n=1:Total_No_of_Objects
    % Normalizing all the objects for removing lightening effects
    % on them then increasing the resolution of object detection
    % and recognition. Note that, the normalization will be done
    % just for the faces pixels for keeping variations among the
    % objects and impulses just in the faces without the
    % backgrounds effects.
    Object_Number=[num2str(n) '.jpg'];
    Object_Location=fullfile(Objects_Folder,Object_Number);
    Object1=double(rgb2gray(imread(Object_Location))); % The object.
    T=Object1>Objects_Threshold; % The pixels bigger than the
    % threshold are of interest
    % because they represent the % pixels of a face.
    % T=Object1<Objects_Threshold; % The pixels smaller than the
    % % threshold are of interest % % because they represent the
    % % pixels of a face.
    Object=zeros(size(T,1),size(T,2)); % The normalized object.
    for R2=1:size(T,1)
        for C2=1:size(T,2)
            if T(R2,C2)==1
                Object(R2,C2)=ceil(255*(Object1(R2,C2)/
                max(max(Object1)))); % The normalization of the
                % object. This is done to % increase the dynamic
                % range of the object for % visualization by scaling
                % the intensities from 0 % to 255.
        end
        end
    end
    [r c]=size(Object);
    % figure
    % subplot(2,1,1)
    % imshow(Object1)
    % if n<=L1
    %    title({['This Is To Show How Good the Objects '.....
    % 'Threshold Is,'';blanks(1);'Object No.' ....
    % num2str(n) ' for Mr. Mansour Alshammari']})
elseif n>L1 && n<=L2
    title(["This Is To Show How Good the Objects ", ...
           'Threshold Is,'],blanks(1);['Object No. ', ....
           num2str(n) ' for Mr. Methkir Alharthee'])
elseif n>L2 && n<=L3
    title(["This Is To Show How Good the Objects ", ...
           'Threshold Is,'],blanks(1);['Object No. ', ....
           num2str(n) ' for Mr. Mohammed Hanafy'])
end

subplot(2,1,2)
imshow(Object)
if n<=L1
    title(["Normalized Object No. ' num2str(n) ....
           ' for Mr. Mansour Alshammari'])
elseif n>L1 && n<=L2
    title(["Normalized Object No. ' num2str(n) ....
           ' for Mr. Methkir Alharthee'])
elseif n>L2 && n<=L3
    title(["Normalized Object No. ' num2str(n) ....
           ' for Mr. Mohammed Hanafy'])
end

figure
subplot(2,1,1)
imshow(uint8(Object1))
if n<=L1
    title(["Object No. ' num2str(n) ....
           ' for Mr. Mansour Alshammari'])
elseif n>L1 && n<=L2
    title(["Object No. ' num2str(n) ....
           ' for Mr. Methkir Alharthee'])
elseif n>L2 && n<=L3
    title(["Object No. ' num2str(n) ....
           ' for Mr. Mohammed Hanafy'])
end
subplot(2,1,2)
imshow(uint8(Object))
if n<=L1
    title(["Normalized Object No. ' num2str(n) ....
           ' for Mr. Mansour Alshammari'])
elseif n>L1 && n<=L2
    title(["Normalized Object No. ' num2str(n) ....
           ' for Mr. Methkir Alharthee'])
elseif n>L2 && n<=L3
    title(["Normalized Object No. ' num2str(n) ....
           ' for Mr. Mohammed Hanafy'])
end
for m=1:Total_No_of_Impulses
% Normalizing all the impulses for removing lightening effects on them then increasing the resolution of object detection and recognition. Note that, the normalization will be done just for the faces pixels for keeping variations among the objects and impulses just in the faces without the backgrounds effects.

One_Impulse=Impulses(m).name;
Impulse_Location=fullfile(Impulses_Folder,One_Impulse);
Impulse1=double(rgb2gray(...)); % The impulse response.

T1=Impulse1>Impulses_Threshold; % The pixels bigger than the threshold are of interest because they represent the pixels of a face.

T1=Impulse1<Impulses_Threshold; % The pixels smaller than the threshold are of interest because they represent the pixels of a face.

Impulse=zeros(size(T1,1),size(T1,2)); % The normalized impulse response.

for R1=1:size(T1,1)
    for C1=1:size(T1,2)
        if T1(R1,C1)==1
            Impulse(R1,C1)=ceil(255*(Impulse1(R1,C1)/max(max(Impulse1)))); % The normalization of the impulse response. This is done to increase the dynamic range of the impulse for visualization by scaling the intensities from 0 to 255.
        end
    end
end

[p q]=size(Impulse);

% Equalizing the width of the impulse response with the width of the object in order to collimate them on the input transparencies.
if q>c
    if mod(q-c,2)==0
        Object=[zeros(r,(q-c)/2) Object zeros(r,(q-c)/2)];
    elseif mod(q-c,2)==1

Object=[zeros(r,floor((q-c)/2)) Object .... 
    zeros(r,floor((q-c)/2)+1)];
end

elseif q<c
    if mod(c-q,2)==0
        Impulse=[zeros(p,(c-q)/2) Impulse ..... 
            zeros(p,(c-q)/2)];
    elseif mod(c-q,2)==1
        Impulse=[zeros(p,floor((c-q)/2)) Impulse ..... 
            zeros(p,floor((c-q)/2)+1)];
    end
end

% Collimating the impulse response and the object on the 
% input transparencies. Note that, the separation between 
% the impulse response and object must be bigger than 
% max{Wh,Wg}+Wg/2+Wh/2 in order to make the output 
% crosscorrelations of the impulse response and object are 
% completely separated without any overlapping.
Wh=size(Impulse,1); % The width of the impulse response in 
% the direction of the y1-coordinate.
Wg=size(Object,1); % The width of the object in the 
% direction of the y1-coordinate.
Y=max(Wh,Wg)+((Wh+Wg)/2); % The separation between the 
% centers of the impulse response 
% and object.
dis=10; % This distance in order to make the separation 
% between the centers of the impulse response and 
% object bigger than max{Wh,Wg}+Wg/2+Wh/2.
Impulse_Object=[........ 
    zeros(ceil(max(Wh,Wg)/2+Wh/4+3/4*Wg)+ceil(dis/2),..... 
    size(Impulse,2));Impulse;..... 
    zeros(max(Wh,Wg)+dis,size(Impulse,2));Object;..... 
    zeros(ceil(max(Wh,Wg)/2+3/4*Wh+Wg/4)+ceil(dis/2),..... 
    size(Object,2))];

% Equalizing the dimensions of the input plane P1 by 
% padding it with zeros.
[R C]=size(Impulse_Object);
if R>C
    if mod(R-C,2)==0
        Impulse_Object=[........ 
            zeros(size(Impulse_Object,1),(R-C)/2) ..... 
            Impulse_Object ......
            zeros(size(Impulse_Object,1),(R-C)/2)];
    elseif mod(R-C,2)==1
        Impulse_Object=[........ 
            zeros(size(Impulse_Object,1),..... 
            floor((R-C)/2)) Impulse_Object ......
            zeros(size(Impulse_Object,1),.....]
floor((R-C)/2)+1));

    end
elseif R<C
    if mod(C-R,2)==0
        Impulse_Object=[......
        zeros((C-R)/2,size(Impulse_Object,2));....
        Impulse_Object;....
        zeros((C-R)/2,size(Impulse_Object,2))];
    elseif mod(C-R,2)==1
        Impulse_Object=[zeros(floor((C-R)/2),......
        size(Impulse_Object,2));Impulse_Object;......
        zeros(floor((C-R)/2)+1,......
        size(Impulse_Object,2))];
    end
end

U1=Impulse_Object; % The transmitted field from
% the input plane P1.
[M N]=size(U1);

L=10; % The physical side length of the
% array which holds the input
% plane P1 in meters (m).
dx1_Input=L/N; % The sample spacing in the input plane
% array in the direction of the spatial
% space coordinate x1 in meters (m).
dy1_Input=L/M; % The sample spacing in the input plane
% array in the direction of the spatial
% space coordinate y1 in meters (m).
x1_Axis_Input=floor(N/2)*dx1_Input:dx1_Input:....
    ceil(N/2)*dx1_Input-dx1_Input; % Sampling the input
    % plane P1 in the % direction of the % spatial % coordinate x1.
y1_Axis_Input=floor(M/2)*dy1_Input:dy1_Input:....
    ceil(M/2)*dy1_Input-dy1_Input; % Sampling the input
    % plane P1 in the % direction of the % spatial % coordinate y1.

U2=fftshift(fft2(fftshift(U1))); % The Fourier transform of
% the transmitted field in % the back focal plane of % the lens L2.
I=(abs(U2)).^2; % The intensity of the Fourier transformed % field in the plane P2.
lambda=550e-9; % The wavelength in meters (m).
f=0.055; % The focal length in meters (m).
dx2=(lambda*f)/(N*dx1_Input); % The sample spacing in the % plane P2 in the direction % of the spatial space % coordinate x2 in meters % (m).
dy2=(lambda*f)/(M*dy1_Input); % The sample spacing in the % plane P2 in the direction % of the spatial space % coordinate y2 in meters % (m).
x2_Axis=floor(N/2)*dx2:dx2:........ ceil(N/2)*dx2-dx2; % Sampling the plane P2 in the % direction of the spatial space % coordinate x2.
y2_Axis=floor(M/2)*dy2:dy2:..... ceil(M/2)*dy2-dy2; % Sampling the plane P2 in the % direction of the spatial space % coordinate y2.

U3=ifftshift(ifft2(ifftshift(I))); % The crosscorrelated % field in the back % focal plane of the % lens L4.

dx3=(lambda*f)/(N*dx2); % The sample spacing in the plane % P3 in the direction of the % spatial space coordinate x3 in % meters (m).
dy3=(lambda*f)/(M*dy2); % The sample spacing in the plane % P3 in the direction of the % spatial space coordinate y3 in % meters (m).
x3_Axis=floor(N/2)*dx3:dx3:..... ceil(N/2)*dx3-dx3; % Sampling the plane P3 in the % direction of the spatial space % coordinate x3.
y3_Axis=floor(M/2)*dy3:dy3:..... ceil(M/2)*dy3-dy3; % Sampling the plane P3 in the % direction of the spatial space % coordinate y3.

% Synthesizing a desired filtering mask then filtering the % crosscorrelated field in the plane P3.
Cen=floor(M/2)+1; % The center of the filtering mask.
Cen1=Cen-(Y+dis); % The center of the desired
% crosscorrelated field.
Wh1=q; % The width of the impulse response in
% the direction of the x1-coordinate.
Wg1=c; % The width of the object in the
% direction of the x1-coordinate.

Mask=zeros(M,N);
for P=Cen1-floor((Wh+Wg)/2):Cen1+ceil((Wh+Wg)/2)
    for Q=Cen-floor((Wh1+Wg1)/2):Cen+ceil((Wh1+Wg1)/2)
        Mask(P,Q)=1;
    end
end

Cross_Field=Mask.*U3; % The filtered crosscorrelated
% field in the plane P3.

% For simplicity, instead of processing the entire image of
% the filtered crosscorrelated field, we select only the
% crosscorrelated field of interest.
P=Cen1-floor((Wh+Wg)/2):Cen1+ceil((Wh+Wg)/2);
Q=Cen-floor((Wh1+Wg1)/2):Cen+ceil((Wh1+Wg1)/2);
Desired_Cross_Field=U3(P,Q);

Max_Desired_Cross_Fields(n,m)=max(max(Desired_Cross_Field));

figure
subplot(2,1,1)
imshow(Impulse1)
if m==1
    title({['This Is To Show How Good the Impulses '.....
    'Threshold Is,']; blanks(1);.....
    ['Impulse Response No.' num2str(m) .....}
    ' for Mr. Mansour Alshammari']})
elseif m==2
    title({['This Is To Show How Good the Impulses '.....
    'Threshold Is,'];blanks(1);.....
    ['Impulse Response No.' num2str(m) .....}
    ' for Mr. Methkir Alharthee']})
elseif m==3
    title({['This Is To Show How Good the Impulses '....
    'Threshold Is,'];blanks(1);.....
    ['Impulse Response No.' num2str(m) .....}
    ' for Mr. Mohammed Hanafy']})
end
subplot(2,1,2)
% imshow(Impulse)
if m==1
    title({['Normalized Impulse Response No.' ....
        num2str(m) ' for Mr. Mansour Alshammari']}))
elseif m==2
    title({['Normalized Impulse Response No.' ....
        num2str(m) ' for Mr. Methkir Alharthee']}))
else m==3
    title({['Normalized Impulse Response No.' ....
        num2str(m) ' for Mr. Mohammed Hanafy']}))
end

figure
subplot(2,1,1)
imshow(uint8(Impulse1))
if m==1
    title({['Impulse Response No.' num2str(m) ....
        ' for Mr. Mansour Alshammari']}))
elseif m==2
    title({['Impulse Response No.' num2str(m) ....
        ' for Mr. Methkir Alharthee']}))
elseif m==3
    title({['Impulse Response No.' num2str(m) ....
        ' for Mr. Mohammed Hanafy']}))
end
subplot(2,1,2)
imshow(uint8(Impulse))
if m==1
    title({['Normalized Impulse Response No.' ....
        num2str(m) ' for Mr. Mansour Alshammari']}))
elseif m==2
    title({['Normalized Impulse Response No.' ....
        num2str(m) ' for Mr. Methkir Alharthee']}))
elseif m==3
    title({['Normalized Impulse Response No.' ....
        num2str(m) ' for Mr. Mohammed Hanafy']}))
end

figure('units','centimeters','position',[7 1.2 25 16.9])
imagesc(xl_Axis_Input,yl_Axis_input,U1)
colorbar
if n<=L1
    if m==1
        title({['The Transmitted Field from the ' ....
            'Input Plane P_1'];['(Impulse No.1 Is ' ....
            'for Mr. Mansour Alshammari as Well ' ....
            'as Object No.' num2str(n) ....
            ' Is for Him)']}))
    elseif m==2
        title({['The Transmitted Field from the ' ....
            'Input Plane P_2'];['(Impulse No.2 Is ' ....
            'for Mr. Methkir Alharthee as Well ' ....
            'as Object No.' num2str(n) ....
            ' Is for Him)']}))
    elseif m==3
        title({['The Transmitted Field from the ' ....
            'Input Plane P_3'];['(Impulse No.3 Is ' ....
            'for Mr. Mohammed Hanafy as Well ' ....
            'as Object No.' num2str(n) ....
            ' Is for Him)']}))
end
Appendix F. A Code for the Optical Model

% 'Input Plane P_1'; ['(Impulse No.2 Is '....
% 'for Mr. Methkir Alharthee and Object '....
% 'No.' num2str(n) .......
% ' Is for Mr. Mansour Alshammari')]})
else
    title({['The Transmitted Field from the '.....
% 'Input Plane P_1'; ['(Impulse No.3 Is '....
% 'for Mr. Mohammed Hanafy and Object '....
% 'No.' num2str(n) .......
% ' Is for Mr. Mansour Alshammari')]})
end
% elseif n>L1 && n<=L2
    if m==1
        title({['The Transmitted Field from the '.....
% 'Input Plane P_1'; ['(Impulse No.1 Is '....
% 'for Mr. Mansour Alshammari and Object '....
% 'No.' num2str(n) .......
% ' Is for Mr. Methkir Alharthee')]})
    elseif m==2
        title({['The Transmitted Field from the '.....
% 'Input Plane P_1'; ['(Impulse No.2 Is '....
% 'for Mr. Methkir Alharthee as Well as '....
% 'Object No.' num2str(n) ' Is for Mr. Mohammed Hanafy)']})
    else
        title({['The Transmitted Field from the '.....
% 'Input Plane P_1'; ['(Impulse No.3 Is '....
% 'for Mr. Mohammed Hanafy as Well as '....
% 'Object No.' num2str(n) ' Is for Mr. Methkir Alharthee)']})
    end
% elseif n>L2 && n<=L3
    if m==1
        title({['The Transmitted Field from the '.....
% 'Input Plane P_1'; ['(Impulse No.1 Is '....
% 'for Mr. Mansour Alshammari and Object '....
% 'No.' num2str(n) .......
% ' Is for Mr. Mohammed Hanafy)']})
    elseif m==2
        title({['The Transmitted Field from the '.....
% 'Input Plane P_1'; ['(Impulse No.2 Is '....
% 'for Mr. Methkir Alharthee and '.....
% 'Object No.' num2str(n) .......
% ' Is for Mr. Mohammed Hanafy)']})
    else
        title({['The Transmitted Field from the '.....
% 'Input Plane P_1'; ['(Impulse No.3 Is '....
% 'for Mr. Mohammed Hanafy as Well as '.....
% 'Object No.' num2str(n) ' Is for Him)']})
    end
% end
end
 colormap('gray')
  xlabel('x_1 (m)')
  ylabel('y_1 (m)')

  figure('units','centimeters','position',[7 1.2 25 16.9])
  imagesc(x2_Axis,y2_Axis,255*(I/max(max(I))))
  colorbar

  if n<=L1
    if m==1
      title({['The Incident Intensity on the '.....
        'Plane P_2';:['Impulse No.1 Is for '.....
        'Mr. Mansour Alshammari as Well as '.....
        'Object No.' num2str(n) ' Is for Him')])
    elseif m==2
      title({['The Incident Intensity on the '.....
        'Plane P_2';:['Impulse No.2 Is for '.....
        'Mr. Methkir Alharthee and Object No.' ....
        num2str(n) ' Is for Mr. Mansour Alshammari']})
    else
      title({['The Incident Intensity on the '.....
        'Plane P_2';:['Impulse No.3 Is for '.....
        'Mr. Mohammed Hanafy and Object No.' ....
        num2str(n) ' Is for Mr. Methkir Alharthee']})
  end

  elseif n>L1 && n<=L2
    if m==1
      title({['The Incident Intensity on the '.....
        'Plane P_2';:['Impulse No.1 Is for '.....
        'Mr. Mansour Alshammari and Object No.' ....
        num2str(n) ' Is for Mr. Methkir Alharthee']})
    elseif m==2
      title({['The Incident Intensity on the '.....
        'Plane P_2';:['Impulse No.2 Is for '.....
        'Mr. Methkir Alharthee as Well as '.....
        'Object No.' num2str(n) ' Is for Him']})
    else
      title({['The Incident Intensity on the '.....
        'Plane P_2';:['Impulse No.3 Is for '.....
        'Mr. Mohammed Hanafy and '.....
        'Object No.' num2str(n) ' Is for Mr. Methkir Alharthee']})
  end

  elseif n>L2 && n<=L3
    if m==1
      title({['The Incident Intensity on the '.....
        'Plane P_2';:['Impulse No.1 Is for '.....
        'Mr. Mansour Alshammari and '.....

Appendix F. A Code for the Optical Model

    % 'Object No.' num2str(n) ......
    % ' Is for Mr. Mohammed Hanafy])
    elseif m==2
        title({['The Incident Intensity on the '....
                'Plane P_2'; '
                'Impulse No.2 Is for '....
                'Mr. Methkir Alharthee and '....
                'Object No.' num2str(n) ......
                ' Is for Mr. Mohammed Hanafy'])
    else
        title({['The Incident Intensity on the '.....
                'Plane P_2'; '
                'Impulse No.3 Is for '....
                'Mr. Mohammed Hanafy as Well as '....
                'Object No.' num2str(n) ' Is for Him'])
    end

    colormap('gray')
    xlabel('x_2 (m)')
    ylabel('y_2 (m)')
    figure('units','centimeters','position',[7 1.2 25 16.9])
    imagesc(x3_Axis,y3_Axis,U3)
    colorbar
    if n<=L1
        if m==1
            title({['The Crosscorrelated Field in '.....
                    'the Plane P_3'; '
                    'Impulse No.1 Is '.....
                    'for Mr. Mansour Alshammari as Well as '....
                    'Object No.' num2str(n) ' Is for Him'])
        elseif m==2
            title({['The Crosscorrelated Field in '....
                    'the Plane P_3'; '
                    'Impulse No.2 Is '....
                    'for Mr. Methkir Alharthee and '....
                    'Object No.' num2str(n) ......
                    ' Is for Mr. Mansour Alshammari'])}
        else
            title({['The Crosscorrelated Field in '.....
                    'the Plane P_3'; '
                    'Impulse No.3 Is '.....
                    'for Mr. Mohammed Hanafy and '....
                    'Object No.' num2str(n) ......
                    ' Is for Mr. Mansour Alshammari'])}
        end
    elseif n>L1 && n<=L2
        if m==1
            title({['The Crosscorrelated Field in '.....
                    'the Plane P_3'; '
                    'Impulse No.1 Is '.....
                    'for Mr. Mansour Alshammari and '....
                    'Object No.' num2str(n) ......
                    ' Is for Mr. Methkir Alharthee'])}
        elseif m==2
            title({['The Crosscorrelated Field in '....
                    'the Plane P_3'; '
                    'Impulse No.2 Is '....
                    'for Mr. Mohammed Hanafy and '....
                    'Object No.' num2str(n) ......
                    ' Is for Mr. Methkir Alharthee'])}
        else
            title({['The Crosscorrelated Field in '.....
                    'the Plane P_3'; '
                    'Impulse No.3 Is '....
                    'for Mr. Mohammed Hanafy and '....
                    'Object No.' num2str(n) ......
                    ' Is for Mr. Methkir Alharthee'])}
        end
    else
        title({['The Crosscorrelated Field in '.....
                'the Plane P_3'; '
                'Impulse No.1 Is '.....
                'for Mr. Mansour Alshammari and '....
                'Object No.' num2str(n) ......
                ' Is for Mr. Methkir Alharthee'])}
    else m==2
        title({['The Crosscorrelated Field in '.....
                'the Plane P_3'; '
                'Impulse No.2 Is '....
                'for Mr. Mohammed Hanafy and '....
                'Object No.' num2str(n) ......
                ' Is for Mr. Methkir Alharthee'])}
    end
    end

% colormap('gray')
% xlabel('x_2 (m)')
% ylabel('y_2 (m)')
% figure('units','centimeters','position',[7 1.2 25 16.9])
% title('The Crosscorrelated Field in the Plane P_3')
% title('Impulse No.1 Is for Mr. Mansour Alshammari as Well as Object No. num2str(n) Is for Him')
% colorbar
% 'the Plane P_3';(['Impulse No.2 Is '....
% 'for Mr. Methkhir Alharthee as Well as '....
% 'Object No.' num2str(n) ' Is for Him']})

else
    title({['The Crosscorrelated Field in '.....
% 'the Plane P_3';(['Impulse No.3 Is '....
% 'for Mr. Mohammed Hanafy and '.....
% 'Object No.' num2str(n) ' Is for Mr. Methkir Alharthee']})
end

elseif n>L2 && n<=L3
    if m==1
        title({['The Crosscorrelated Field in '.....
% 'the Plane P_3';(['Impulse No.1 Is '....
% 'for Mr. Mansour Alshammari and '.....
% 'Object No.' num2str(n) ' Is for Mr. Mohammed Hanafy']})
    elseif m==2
        title({['The Crosscorrelated Field in '.....
% 'the Plane P_3';(['Impulse No.2 Is '....
% 'for Mr. Methkhir Alharthee and '.....
% 'Object No.' num2str(n) ' Is for Mr. Mohammed Hanafy']})
    else
        title({['The Crosscorrelated Field in '.....
% 'the Plane P_3';(['Impulse No.3 Is '....
% 'for Mr. Mohammed Hanafy as Well as '....
% 'Object No.' num2str(n) ' Is for Him']})
end

colormap('gray')
xlabel('x_3 (m)')
ylabel('y_3 (m)')

figure('units','centimeters','position',[7 1.2 25 16.9])
imagesc(x3_Axis,y3_Axis,Mask)
colorbar

colorbar

if n<=L1
    if m==1
        title({'The Adaptive Filtering Mask';.....
% ['(Impulse No.1 Is for Mr. Mansour'.....
% 'Alshammari as Well as Object No.'.....
% num2str(n) ' Is for Him']})
    elseif m==2
        title({'The Adaptive Filtering Mask';.....
% ['(Impulse No.2 Is for Mr. Methkhir'.....
% 'Alharthee and Object No.' num2str(n) '....
% ' Is for Mr. Mansour Alshammari]})
    elseif m==3
        title({'The Adaptive Filtering Mask';.....
% ['(Impulse No.3 Is for Mr. Methkhir'.....
% 'Alharthee and Object No.' num2str(n) '....
% ' Is for Mr. Mohammed Hanafy]})
    else
        title({'The Adaptive Filtering Mask';.....
% ['(Impulse No.4 Is for Mr. Mohammed'.....
% 'Hanafy as Well as Object No.' num2str(n) '....
% ' Is for Mr. Mansour Alshammari]})
end

colorbar

if n>L2 && n<=L3
    if m==1
        title({'The Adaptive Filtering Mask';.....
% ['(Impulse No.1 Is for Mr. Mansour'.....
% 'Alshammari as Well as Object No.'.....
% num2str(n) ' Is for Him']})
    elseif m==2
        title({'The Adaptive Filtering Mask';.....
% ['(Impulse No.2 Is for Mr. Methkhir'.....
% 'Alharthee and Object No.' num2str(n) '....
% ' Is for Mr. Mohammed Hanafy]})
    else
        title({'The Adaptive Filtering Mask';.....
% ['(Impulse No.3 Is for Mr. Methkhir'.....
% 'Alharthee and Object No.' num2str(n) '....
% ' Is for Mr. Mohammed Hanafy]})
end

colorbar

if n>L3 && n<=L4
    if m==1
        title({'The Adaptive Filtering Mask';.....
% ['(Impulse No.1 Is for Mr. Mansour'.....
% 'Alshammari as Well as Object No.'.....
% num2str(n) ' Is for Him']})
    elseif m==2
        title({'The Adaptive Filtering Mask';.....
% ['(Impulse No.2 Is for Mr. Methkhir'.....
% 'Alharthee and Object No.' num2str(n) '....
% ' Is for Mr. Mohammed Hanafy]})
    else
        title({'The Adaptive Filtering Mask';.....
% ['(Impulse No.3 Is for Mr. Methkhir'.....
% 'Alharthee and Object No.' num2str(n) '....
% ' Is for Mr. Mohammed Hanafy]})
end

colorbar
Appendix F. A Code for the Optical Model

```matlab
830  %  
831  %    ['(Impulse No.3 Is for Mr. Mohammed Hanafy and Object No.' num2str(n) .....  
832  %     ' Is for Mr. Mansour Alshammari')]})  
833  % end  
834  % elseif n>L1 && n<=L2  
835  % if m==1  
836  %   title({'The Adaptive Filtering Mask';.....  
837  %     ['(Impulse No.1 Is for Mr. Mansour .....  
838  %     'Alshammari and Object No.' num2str(n) .....  
839  %     ' Is for Mr. Methkir Alharthee')]})  
840  % elseif m==2  
841  %   title({'The Adaptive Filtering Mask';.....  
842  %     ['(Impulse No.2 Is for Mr. Methkir .....  
843  %     'Alharthee as Well as Object No.' ....  
844  %     num2str(n) ' Is for Him')]})  
845  % else  
846  %   title({'The Adaptive Filtering Mask';.....  
847  %     ['(Impulse No.3 Is for Mr. Mohammed '....  
848  %     'Hanafy and Object No.' num2str(n) .....  
849  %     ' Is for Mr. Methkir Alharthee')]})  
850  % end  
851  % elseif n>L2 && n<=L3  
852  % if m==1  
853  %   title({'The Adaptive Filtering Mask';.....  
854  %     ['(Impulse No.1 Is for Mr. Mansour .....  
855  %     'Alshammari and Object No.' num2str(n) .....  
856  %     ' Is for Mr. Mohammed Hanafy')]})  
857  % elseif m==2  
858  %   title({'The Adaptive Filtering Mask';.....  
859  %     ['(Impulse No.2 Is for Mr. Methkir .....  
860  %     'Alharthee and Object No.' num2str(n) .....  
861  %     ' Is for Mr. Mohammed Hanafy')]})  
862  % else  
863  %   title({'The Adaptive Filtering Mask';.....  
864  %     ['(Impulse No.3 Is for Mr. Mohammed '....  
865  %     'Hanafy as Well as Object No.' .....  
866  %     num2str(n) ' Is for Him')]})  
867  % end  
868  % end  
869  % colormap('gray')  
870  % xlabel('x_3 (m)')  
871  % ylabel('y_3 (m)')  
872  % figure('units','centimeters','position',[7 1.2 25 16.9])  
873  % imagesc(x3_Axis,y3_Axis,Cross_Field)  
874  % colorbar  
875  % if n<=L1  
876  %    if m==1  
877  %      title({['The Filtered Crosscorrelated '....  
878  %        'Field in the Plane P_3'];['(Impulse '....
```
elseif m==2
    title({['The Filtered Crosscorrelated '...
            'Field in the Plane P_3'];
            ['(Impulse '...
            'No.2 Is for Mr. Methkir Alharthee '...
            'and Object No.' num2str(n) .......
            ' Is for Mr. Mansour Alshammari')]})
elseif n>L1 && n<=L2
    if m==1
        title({['The Filtered Crosscorrelated '...
                'Field in the Plane P_3'];
                ['(Impulse '...
                'No.1 Is for Mr. Mansour Alshammari '...
                'and Object No.' num2str(n) .......
                ' Is for Mr. Methkir Alharthee')]})
    elseif m==2
        title({['The Filtered Crosscorrelated '...
                'Field in the Plane P_3'];
                ['(Impulse '...
                'No.2 Is for Mr. Methkir Alharthee '...
                'as Well as Object No.' num2str(n) ....
                ' Is for Mr. Mansour Alshammari')]})
    else
        title({['The Filtered Crosscorrelated '...
                'Field in the Plane P_3'];
                ['(Impulse '...
                'No.3 Is for Mr. Mohammed Hanafy '...
                'Object No.' num2str(n) ....
                ' Is for Mr. Methkir Alharthee')]})
    end
elseif n>L2 && n<=L3
    if m==1
        title({['The Filtered Crosscorrelated '...
                'Field in the Plane P_3'];
                ['(Impulse '...
                'No.1 Is for Mr. Mansour Alshammari '...
                'and Object No.' num2str(n) ....
                ' Is for Mr. Mohammed Hanafy')]})
    elseif m==2
        title({['The Filtered Crosscorrelated '...
                'Field in the Plane P_3'];
                ['(Impulse '...
                'No.2 Is for Mr. Methkir Alharthee '...
                'and Object No.' num2str(n) ....
                ' Is for Mr. Mohammed Hanafy')]})
    else
        title({['The Filtered Crosscorrelated '...
                'Field in the Plane P_3'];
                ['(Impulse '...
                'No.3 Is for Mr. Mohammed Hanafy '...
                'Object No.' num2str(n) ....
                ' Is for Mr. Methkir Alharthee')]})
    end
elseif n>L3
    if m==1
        title({['The Filtered Crosscorrelated '...
                'Field in the Plane P_3'];
                ['(Impulse '...
                'No.1 Is for Mr. Mansour Alshammari '...
                'and Object No.' num2str(n) ....
                ' Is for Mr. Mohammed Hanafy')]})
    elseif m==2
        title({['The Filtered Crosscorrelated '...
                'Field in the Plane P_3'];
                ['(Impulse '...
                'No.2 Is for Mr. Methkir Alharthee '...
                'and Object No.' num2str(n) ....
                ' Is for Mr. Mohammed Hanafy')]})
    else
        title({['The Filtered Crosscorrelated '...
                'Field in the Plane P_3'];
                ['(Impulse '...
                'No.3 Is for Mr. Mohammed Hanafy '...
                'Object No.' num2str(n) ....
                ' Is for Mr. Methkir Alharthee')]})
    end
end
% title({['The Filtered Crosscorrelated ' ....
931  ' Field in the Plane P_3'];['(Impulse ' ....
932  ' No.3 Is for Mr. Mohammed Hanafy as ' ....
933  ' Well as Object No.' num2str(n) ' ....
934  ' Is for Him)']})
935  end
936  end
937  colormap('gray')
938  xlabel('x_3 (m)')
939  ylabel('y_3 (m)')
940  figure('units','centimeters','position',[7 1.2 25 16.9])
941  imagesc(x3_Axis,y3_Axis,Desired_Cross_Field)
942  colorbar
943  if n<=L1
944    if m==1
945      title({['The Crosscorrelated Field of ' ....
946        ' Interest in the Plane P_3'];['(Impulse' ....
947        ' No.1 Is for Mr. Mansour ' ....
948        ' Alshammari as Well as Object No.' ....
949        ' Is for Him)']})
950    elseif m==2
951      title({['The Crosscorrelated Field of ' ....
952        ' Interest in the Plane P_3'];['(Impulse' ....
953        ' No.2 Is for Mr. Methkir Alharthee ' ....
954        ' and Object No.' num2str(n) ' ....
955        ' Is for Mr. Mansour Alshammari)']})
956    else
957      title({['The Crosscorrelated Field of ' ....
958        ' Interest in the Plane P_3'];['(Impulse' ....
959        ' No.3 Is for Mr. Mohammed Hanafy and ' ....
960        ' Object No.' num2str(n) ' ....
961        ' Is for Mr. Mansour Alshammari)']})
962    end
963  elseif n>L1 && n<=L2
964    if m==1
965      title({['The Crosscorrelated Field of ' ....
966        ' Interest in the Plane P_3'];['(Impulse' ....
967        ' No.1 Is for Mr. Mansour Alshammari ' ....
968        ' and Object No.' num2str(n) ' ....
969        ' Is for Mr. Methkir Alharthee)']})
970    elseif m==2
971      title({['The Crosscorrelated Field of ' ....
972        ' Interest in the Plane P_3'];['(Impulse' ....
973        ' No.2 Is for Mr. Methkir Alharthee ' ....
974        ' as Well as Object No.' num2str(n) ' ....
975        ' Is for Him)']})
976    else
977      title({['The Crosscorrelated Field of ' ....
978        ' Interest in the Plane P_3'];['(Impulse' ....

\begin{verbatim}
980   \% ' No.3 Is for Mr. Mohammed Hanafy and '....
981   \% 'Object No.' num2str(n) ..... 
982   \% ' Is for Mr. Methkir Alharthee')])}
983   \% end
984   \% elseif n>L2 && n<=L3
985   \% if m==1
986   \% title({['The Crosscorrelated Field of '....
987   \% 'Interest in the Plane P_3'];['(Impulse'....
988   \% ' No.1 Is for Mr. Mansour Alshammari '....
989   \% 'and Object No.' num2str(n) ......
990   \% ' Is for Mr. Mohammed Hanafy')])}
991   \% elseif m==2
992   \% title({['The Crosscorrelated Field of '....
993   \% 'Interest in the Plane P_3'];['(Impulse'....
994   \% ' No.2 Is for Mr. Methkir Alharthee '....
995   \% 'and Object No.' num2str(n) ......
996   \% ' Is for Mr. Mohammed Hanafy')])}
997   \% else
998   \% title({['The Crosscorrelated Field of '....
999   \% 'Interest in the Plane P_3'];['(Impulse'....
1000  \% ' No.3 Is for Mr. Mohammed Hanafy '....
1001  \% 'as Well as Object No.' ......
1002  \% num2str(n) ' Is for Him')})
1003  \% end
1004  \% colormap('gray')
1005  \% xlabel('x_3 (m)')
1006  \% ylabel('y_3 (m)')
1007  \% %%%% Note that, "imshow" is better to be used instead
1008  \% %%%% of "imagesc" during designing the adaptive mask
1009  \% %%%% because "imshow" displays the cross-correlations
1010  \% %%%% in the plane P3 clearer than "imagesc"!!
1011  \% disp(['Please, press any keyboard button to explore '....
1012  \% 'the remaining crosscorrelated fields >>>>>>>'])
1013  \% pause
1014  \% clc
1015  \% close all
1016  \% % Testing the process of object recognition.
1017  \% Note that, the objects here are known for us in order to use them
1018  \% for testing the object recognition process as well as setting up
\end{verbatim}
% a recognition threshold.

% The elements in each of the following vectors represent the
% maximum values of the desired crosscorrelated fields between the
% objects and their corresponding impulse. Note that, each impulse
% and its corresponding objects have a same face so the vectors
% will include the biggest crosscorrelations.
V1=(Max_Desired_Cross_Fields(1:L1,1)).';
V2=(Max_Desired_Cross_Fields(L1+1:L2,2)).';
V3=(Max_Desired_Cross_Fields(L2+1:L3,3)).';

% The calculation of the mean and the standard deviation for each
% vector of the biggest crosscorrelations and stacking them in a
% vector for the means and another for the standard deviations.
% This is done for setting up a threshold for object recognition.
V_Mean=[mean(V1);mean(V2);mean(V3)];
V_STD=[std(V1);std(V2);std(V3)];
save('Computed Means for Object Recognition','V_Mean')
save('Computed STDs for Object Recognition','V_STD')

fid=fopen('Object Recognition Results for Testing.txt',....
'w'); % A text file for typing
% the object recognition
% results for testing.
fprintf(fid,'
***** The Object Recognition Results '.....
'for Testing Obtained from the Code of the *****

Testing and Setting up Thresholds ****

The Object No. The Object Is Originally for The Status

============== =========================== =========

 Failures_Vector=zeros(1,Total_No_of_Objects); % A vector for
% counting the number
% of failures in the
% object recognition
% process.
Latex_Matrix=cell(Total_No_of_Objects,4); % This matrix is used
% for creating a table
% in Latex.

for w=1:Total_No_of_Objects
% The object recognition process.
  for ii=1:Total_No_of_Impulses
    if max(Max_Desired_Cross_Fields(w,:))>=......
      Max_Desired_Cross_Fields(w,ii) && ......
      max(Max_Desired_Cross_Fields(w,:))>=......
      (V_Mean(ii,1)-V_STD(ii,1)) && ......
      max(Max_Desired_Cross_Fields(w,:))<=......
\begin{verbatim}
(V_Mean(ii,1)+V_STD(ii,1))
S=transpose(struct2cell(Impulses));
d=sortrows(S,1);
z=d(:,1);
Recognized_As=char(z(ii,1));
Recognized_Object_Location=fullfile(Impulses_Folder,Recognized_As);
Recognized_Object=im2double(rgb2gray(imread(Recognized_Object_Location)));
break
else
    Recognized_As='Unknown Object';
    Recognized_Object=imread('Unknown_Object.jpg');
end
end

% Defining the object.
if w<=L1
    name='Mr. Mansour Alshammari';
elseif L1<w && w<=L2
    name='Mr. Methkir Alharthee';
else name='Mr. Mohammed Hanafy';
end

Str1=strcmp(Recognized_As(1:length(Recognized_As)-6),....
    'Mr. Mansour Alshammari');
Str2=strcmp(Recognized_As(1:length(Recognized_As)-6),....
    'Mr. Methkir Alharthee');
Str3=strcmp(Recognized.As(1:length(Recognized_As)-6),....
    'Mr. Mohammed Hanafy');
F='Success';
if w<=L1 && Str1==0;
    F='Failure';
    Failures_Vector(1,w)=1;
elseif w>L1 && w<=L2 && Str2==0;
    F='Failure';
    Failures_Vector(1,w)=1;
elseif w>L2 && w<=L3 && Str3==0;
    F='Failure';
    Failures_Vector(1,w)=1;
end

% Object_Number=[num2str(w) '.jpg'];
% Object_Location=fullfile(Objects_Folder,Object_Number);
% Object=im2double(rgb2gray(imread(Object_Location)));
% subplot(2,1,1)
% imshow(Object)
% title({['Object No.' num2str(w) ' Is Originally for'];name})
% subplot(2,1,2)
% imshow(Recognized_Object)
\end{verbatim}
% if Str1==1 || Str2==1 || Str3==1
% title({'The Object Is Recognized As';Recognized_As(1:length(Recognized_As)-6)})
% else title({'The Object Is Recognized As' ['an' blanks(1) Recognized_As]})
% end
if Str1==1 || Str2==1 || Str3==1
   fprintf(fid,['%0.3d	 %s',Recognized_As(1:length(Recognized_As)-6),F]);
   Latex_Matrix(w,1:4)={num2str(w) name Recognized_As F};
   else fprintf(fid,['%0.3d	 %s
',Recognized_As F]);
   Latex_Matrix(w,1:4)={num2str(w) name Recognized_As F};
end
% disp({'Please, press Enter button to see another '...
% 'object and its recognition....'})
% pause
% clc
total_number_of_failures=sum(Failures_Vector); % The total number
% of failures in the
% object recognition
% process.
fprintf(fid,['===============================

** The Total Number of Successes: %0.3d out of %0.3d (%3.4f%%)

** The Total Number of Failures: %0.3d out of %0.3d (%3.4f%%)

fclose(fid);
close all
clc
disp({'Please, see the documented results of object '
' recognition in the open'})
disp({'text file then press any keyboard button to '
'resume the code >>>)})
Text='Object Recognition Results for Testing.txt';
open(Text) % Opening the text file which contains
  % the results of object recognition.
pause
clc
open('The_JTC_Testing_and_Setting_up_Thresholds.m')

% Testing the process of object detection.
% Note that, the objects here are known for us in order to use them
% for testing the object detection process as well as setting up a
% detection threshold.
% The calculation of the mean and standard deviation for the
% crosscorrelations between the objects and each impulse response.
% This is done for setting up a threshold for object detection.
Vectorization=reshape(Max_Desired_Cross.Fields,1,...
size(Max_Desired_Cross.Fields,1)......
size(Max_Desired_Cross.Fields,2));
Mean=mean(Vectorization);
STD=std(Vectorization);
save('Computed Mean for Object Detection','Mean')
save('Computed STD for Object Detection','STD')

fid=fopen('Object Detection Results for Testing.txt',.....
 'w'); % A text file for typing
  % the object detection
  % results for testing.
  fprintf(fid,['
 ***** The Object Detection Results '.......
  'for Testing Obtained from the Code of the *****
  'Test and Setting Up Thresholds ****\n\n']);
  fprintf(fid,[' **** Joint Transform Correlator (JTC) for '......
  'Testing and Setting Up Thresholds ****\n\n']);
  fprintf(fid,['The Object No. The Object Originally '.......
  'Is The Detected Object Is The Status\n\n']);
  fprintf(fid,='================= ================ =========='.......
    ' ================ ================ \n');

Failures_Vector1=zeros(1,.......
Total_No_of_Objects); % A vector for counting the number
  % of failures in the object detection
  % process.
Latex_Matrix=cell(Total_No_of_Objects,4); % This matrix is used
  % for creating a table
  % in Latex.
for w=1:Total_No_of_Objects
  % The object detection process.
  if max(Max_Desired_Cross.Fields(w,:))>=(Mean-STD) && ..... max(Max_Desired_Cross.Fields(w,:))<=(Mean+STD)
    Detected_As='a face';
    Detected_Object=imread('A_Face.jpg');
else
    Detected_As='not a face';
    Detected_Object=imread('Not_a_Face.jpg');
end

b='a face'; % The object originally is a face.

Str=strcmp(Detected_As,'a face');
F='Success';
if w<=L3 & Str==0;
    F='Failure';
    Failures_Vector1(1,w)=1;
end

% Object_Number=[num2str(w) '.jpg'];
% Object_Location=fullfile(Objects_Folder,Object_Number);
% Object=im2double(rgb2gray(imread(Object_Location)));
% subplot(2,1,1)
% imshow(Object)
% title(
%   ['Object No.' num2str(w) ' Originally Is'])
% subplot(2,1,2)
% imshow(Detected_Object)
% title('It Is Detected As')

if Str==1
    fprintf(fid,['%0.3d			 %s 	 %s 
',w,b,Detected_As,F]);
    Latex_Matrix(w,1:4)={num2str(w) b Detected_As F};
else fprintf(fid,['%0.3d			 %s 	 %s 
',w,b,Detected_As,F]);
    Latex_Matrix(w,1:4)={num2str(w) b Detected_As F};
end

% disp(['Please, press Enter button to see another '....
%     'object and its detection....'])
% pause
% clc
end

Total_Number_of_Failures1=sum(Failures_Vector1); % The total number
% of failures in
% the object
% detection
% process.

fprintf(fid,[
    '==================================================================
    
    ** The Total Number of Successes: %0.3d out '....
    'of %0.3d (%3.4f%%) \r\n\n',Total_No_of_Objects,....
    Total_Number_of_Failures1,Total_No_of_Objects,....
]);
Appendix F. A Code for the Optical Model

```matlab
(Total_No_of_Objects - Total_Number_of_Failures1)/Total_No_of_Objects*100;
fprintf(fid,['** The Total Number of Failures: %0.3d '.....
'o out of %0.3d (%3.4f%%) \r\n\n'],Total_Number_of_Failures1,.....
(Total_Number_of_Failures1/Total_No_of_Objects)*100);
fclose(fid);
close all
clc
disp(['Please, see the documented results of object '.....
'detection in the open'])
disp(['text file then press any keyboard button to '.....
'resume the code >>>'])
Text='Object Detection Results for Testing.txt';
open(Text) % Opening the text file which contains
% the results of object detection.
pause
clc
open('The_JTC_Testing_and_Setting_up_Thresholds.m')```
### Results of the Optical Recognition

*Table G.1:* The recognition of all 108 objects by using the joint transform correlator (JTC).

<table>
<thead>
<tr>
<th>Object No.</th>
<th>Input Face</th>
<th>Recognized Output Face</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>2</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>3</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>4</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>5</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>6</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>7</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>8</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mohammed Hanafy</td>
<td>Failure</td>
</tr>
<tr>
<td>9</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mohammed Hanafy</td>
<td>Failure</td>
</tr>
<tr>
<td>10</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>11</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>12</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>13</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>14</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>15</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>16</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>17</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>18</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>19</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>20</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>21</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mohammed Hanafy</td>
<td>Failure</td>
</tr>
<tr>
<td>22</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
</tbody>
</table>

*Continued on the next page ...*
<table>
<thead>
<tr>
<th>Object No.</th>
<th>Input Face</th>
<th>Recognized Output Face</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>24</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>25</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mohammed Hanafy</td>
<td>Failure</td>
</tr>
<tr>
<td>26</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mohammed Hanafy</td>
<td>Failure</td>
</tr>
<tr>
<td>27</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mohammed Hanafy</td>
<td>Failure</td>
</tr>
<tr>
<td>28</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>29</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mohammed Hanafy</td>
<td>Failure</td>
</tr>
<tr>
<td>30</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>31</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mohammed Hanafy</td>
<td>Failure</td>
</tr>
<tr>
<td>32</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mohammed Hanafy</td>
<td>Failure</td>
</tr>
<tr>
<td>33</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mohammed Hanafy</td>
<td>Failure</td>
</tr>
<tr>
<td>34</td>
<td>Mr. Mansour Alshammari</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>35</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mohammed Hanafy</td>
<td>Failure</td>
</tr>
<tr>
<td>36</td>
<td>Mr. Mansour Alshammari</td>
<td>Mr. Mohammed Hanafy</td>
<td>Failure</td>
</tr>
<tr>
<td>37</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>38</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>39</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>40</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>41</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>42</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>43</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>44</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>45</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>46</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>47</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>48</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>49</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>50</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>51</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>52</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>53</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>54</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>55</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>56</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>57</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>58</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>59</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>60</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>61</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
</tbody>
</table>

Continued on the next page ...
## Appendix G. Results of the Optical Recognition

<table>
<thead>
<tr>
<th>Object No.</th>
<th>Input Face</th>
<th>Recognized Output Face</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>62</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>63</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>64</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>65</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>66</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>67</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>68</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>69</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>70</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>71</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>72</td>
<td>Mr. Methkir Alharthee</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>73</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>74</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>75</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>76</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>77</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>78</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>79</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>80</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>81</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>82</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>83</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>84</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>85</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>86</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>87</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>88</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>89</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>90</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>91</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>92</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>93</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>94</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>95</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>96</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>97</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>98</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>99</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>100</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
</tbody>
</table>
## Appendix G. Results of the Optical Recognition

<table>
<thead>
<tr>
<th>Object No.</th>
<th>Input Face</th>
<th>Recognized Output Face</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>102</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>103</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>104</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>105</td>
<td>Mr. Mohammed Hanafy</td>
<td>Mr. Mohammed Hanafy</td>
<td>Success</td>
</tr>
<tr>
<td>106</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>107</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
<tr>
<td>108</td>
<td>Mr. Mohammed Hanafy</td>
<td>Unknown Object</td>
<td>Failure</td>
</tr>
</tbody>
</table>

The table end.
Appendix H

Results of the Optical Detection

Table H.1: The detection of all 108 objects by using the joint transform correlator (JTC).

<table>
<thead>
<tr>
<th>Object No.</th>
<th>Input Object</th>
<th>Detected Output Object</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>2</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>3</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>4</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>5</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>6</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>7</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>8</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>9</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>10</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>11</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>12</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>13</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>14</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>15</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>16</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>17</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>18</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>19</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>20</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>21</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>22</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
</tbody>
</table>

Continued on the next page ...
## Appendix H. Results of the Optical Detection

<table>
<thead>
<tr>
<th>Object No.</th>
<th>Input Object</th>
<th>Detected Object</th>
<th>Output Object</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>a face</td>
<td>not a face</td>
<td>Failure</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
<td></td>
</tr>
</tbody>
</table>

*Continued on the next page...*
<table>
<thead>
<tr>
<th>Object No.</th>
<th>Input Object</th>
<th>Detected Object</th>
<th>Output Object</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>62</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>63</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>64</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>65</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>66</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>67</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>68</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>69</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>70</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>71</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>72</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>73</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>74</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>75</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>76</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>77</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>78</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>79</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>80</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>81</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>82</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>83</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>84</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>85</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>86</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>87</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>88</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>89</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>90</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>91</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>92</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>93</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>94</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>95</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>96</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>97</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>98</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>99</td>
<td>a face</td>
<td>not a face</td>
<td>a face</td>
<td>Failure</td>
</tr>
<tr>
<td>100</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
</tbody>
</table>

Continued on the next page ...
Appendix H. Results of the Optical Detection

<table>
<thead>
<tr>
<th>Object No.</th>
<th>Input Object</th>
<th>Detected Object</th>
<th>Output Object</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>102</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>103</td>
<td>a face</td>
<td>not a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>104</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>105</td>
<td>a face</td>
<td>not a face</td>
<td>not a face</td>
<td>Failure</td>
</tr>
<tr>
<td>106</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>107</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
<tr>
<td>108</td>
<td>a face</td>
<td>a face</td>
<td>a face</td>
<td>Success</td>
</tr>
</tbody>
</table>

The table end.
Appendix I. MSEs Plots of Reconstructing Some Training Faces

**Figure I.1:** The plot of the mean squared errors (MSEs) of reconstructing training face number one for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used.
Figure I.2: The plot of the mean squared errors (MSEs) of reconstructing training face number two for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used.
Figure I.3: The plot of the mean squared errors (MSEs) of reconstructing training face number three for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used.
Figure I.4: The plot of the mean squared errors (MSEs) of reconstructing training face number four for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used.
Figure I.5: The plot of the mean squared errors (MSEs) of reconstructing training face number six for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used.
Figure I.6: The plot of the mean squared errors (MSEs) of reconstructing training face number seven for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used.
Figure I.7: The plot of the mean squared errors (MSEs) of reconstructing training face number eight for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used.
Figure I.8: The plot of the mean squared errors (MSEs) of reconstructing training face number nine for different selected eigenfaces compared with resulted mean squared errors when the PCA and IPCA algorithms are used.
A Code for Optimizing the Optical Model Database

This code is for optimizing the impulses database of the joint transform correlator (JTC) for object detection and face recognition.

```matlab
% Optimizing the Impulses' Database of the Joint Transform Correlator (JTC) for Object Detection and Face Recognition.
clc
clear all
close all

Total_No_of_Impulses=3; % The total number of the impulses.

Objects_Folder=[cd ..........\The Black Background Tested Objects']; % The folder of the black background tested objects.

if isdir(Objects_Folder)==0
    Error_Message=sprintf('Error: The following folder does not exist\n%s', Objects_Folder);
warndlg(Error_Message);
```

break
end

Objects=.......  
dir(fullfile(Objects_Folder,'*.jpg')); % Listing the folder of  
\% the black or white  
\% background objects.

Total_No_of_Objects=length(Objects); % The total number  
\% of the objects.

Objs=[36 36 36]; % Each element in this vector represents  
\% the total number of the objects that  
\% are taken for each impulse.

L1=Objs(1); % L1=60 is the total number of the  
\% objects for Mr. Mansour Alshammari.  
L2=L1+Objs(2); % L2=120 is the total number of the  
\% objects for Mr. Methkir Alharthee.  
L3=L2+Objs(3); % L3=180 is the total number of the  
\% objects for Mr. Mohammed Hanafy.

% Imhist for setting up a threshold to work on just the pixels of  
% a face and throwing the background pixels. Imhist calculates the  
% number of pixels in an object that have the same intensity  
% levels. So, if an object has a unified background then the  
% biggest histogram of the intensity levels will be for the  
% backgroud pixels because the total number of pixels that have  
% the same intensity levels are the background pixels of the  
% object. Note that, the histogram of a digital image is defined as  
% the discrete function, h(rk)=nk, where rk is the kth intensity  
% level and nk is the number of pixels in the image whose intensity  
% level is rk.

hist_Objects=zeros(Total_No_of_Objects,256);
for A1=1:Total_No_of_Objects

    Object_Number=[num2str(A1) '.jpg'];
    Object_Location(fullfile(Objects_Folder,Object_Number));
    Object=double(rgb2gray(imread(Object_Location))); % The object.
    hist_Objects(A1,:)=.....
    imhist(uint8(Object)); % Note that, each object must be  
\% scaled between 0 to 255 before  
\% using imhist. For doing that,  
\% uint8 can be used for converting  
\% the object class form double to  
\% uint8.

    if A1<=L1

    plot(hist_Objects(A1,:))
    % plot(hist_Objects(A1,:))
Appendix J. A Code for Optimizing the Optical Model Database

77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
Mean_hist_Objects=sum(hist_Objects,1)/Total_No_of_Objects; % The average histogram for all objects.
% plot(Mean_hist_Objects)
% title('The Mean Histogram of All Objects')
% xlabel('Intensity Level r_(k)')
% ylabel({'The Mean Number of Pixels from All Objects Whose Intensity Level Is r_(k) Where h(r_(k))=n_(k)'})
% axis tight
% pause
% clc
end

% Setting up a threshold in order to work on just the pixels of the faces and blocking the pixels of the backgrounds.
Objects_Threshold=8; % The picked threshold is based on the average histogram for all objects when the objects have black backgrounds. Note that, all intensity levels below the threshold represent the objects backgrounds because these levels have the biggest histogram.
% Objects_Threshold=180; % The Picked threshold is based on the average histogram for all objects when the objects have white backgrounds. Note that, all intensity levels above the threshold represent the objects backgrounds because these levels have the biggest histogram.
Max_Desired_Cross_Fields=zeros(Total_No_of_Objects,Total_No_of_Impulses); % Each element of each row in this
Recognition_Combinations=zeros(L1*(L2-L1)*(L3-L2),......
    Total_No_of_Impulses+1); % The first three columns of this
    % matrix contain different
    % combinations of impulses and the
    % third column represents the error
    % rates of recognition associated with
    % those combinations.

Detection_Combinations=zeros(L1*(L2-L1)*(L3-L2),......
    Total_No_of_Impulses+1); % The first three columns of this
    % matrix contain different
    % combinations of impulses and the
    % third column represents the error
    % rates of detection associated with
    % those combinations.

i=0;

for Im1=1:L1
    for Im2=L1+1:L2
        for Im3=L2+1:L3

        Impulses=[Im1 Im2 Im3]; % The selected impulses.

        for n=1:Total_No_of_Objects

            % Normalizing all the objects for removing
            % lightening effects on them then increasing the
            % resolution of object detection and recognition.
            % Note that, the normalization will be done just
            % for the faces pixels for keeping variations among
            % the objects and impulses just in the faces
            % without the backgrounds effects.
            Object_Number=[num2str(n) '.jpg'];
            Object_Location=.....
            fullfile(Objects_Folder,Object_Number);
            Object1=double(rgb2gray(.....
            imread(Object_Location))); % The object.

            T=Object1>Objects_Threshold; % The pixels bigger
            % than the threshold
% are of interest
% because they
% represent the pixels
% of a face.
T=Object1<Objects_Threshold; % The pixels smaller
% than the threshold
% are of interest
% because they
% represent the
% pixels of a face.
Object=zeros(size(T,1),size(T,2)); % The normalized
% object.

for R2=1:size(T,1)
    for C2=1:size(T,2)
        if T(R2,C2)==1
            Object(R2,C2)=....
            ceil(255*(Object1(R2,C2)/max(....
                max(Object1)))); % The
            % normalization
            % of the object.
            % This is done to
            % increase the
            % dynamic range of
            % the object for
            % visualization by
            % scaling the
            % intensities from
            % 0 to 255.
        end
    end
end
[r c]=size(Object);

for m=1:Total_No_of_Impulses

    Impulse_Number=[num2str(Impulses(m)) '.jpg'];
    Impulse_Location=....
    fullfile(Objects_Folder,Impulse_Number);
    Impulse1=double(rgb2gray(....
        imread(Impulse_Location))); % The impulse
        % response.
T1=Impulse1>......
Objects_Threshold; % The pixels bigger than
% the threshold are of
% interest because they
% represent the pixels
% of a face.

T1=Impulse1<......
Objects_Threshold; % The pixels smaller
% than the threshold
% are of interest
% because they
% represent the
% pixels of a face.

Impulse=zeros(size(T1,1),...
size(T1,2)); % The normalized
% impulse response.
for R1=1:size(T1,1)
 for C1=1:size(T1,2)
 if T1(R1,C1)==1
 Impulse(R1,C1)=.....
 ceil(255*(Impulse1(R1,C1)/...
 max(max(Impulse1))))....
 end
 end
[p q]=size(Impulse);

% Equalizing the width of the impulse response
% with the width of the object in order to
% collimate them on the input transparencies.
if q>c
 if mod(q-c,2)==0
 Object=[zeros(r,(q-c)/2) Object ......
 zeros(r,(q-c)/2)];
 elseif mod(q-c,2)==1
 Object=[zeros(r,floor((q-c)/2)) .....
 Object zeros(r,floor((q-c)/2)+1)];
 end
 elseif q<c
 if mod(c-q,2)==0
 Impulse=[zeros(p,(c-q)/2) Impulse ....
 zeros(p,(c-q)/2)];
 else
 Object=[zeros(p,(c-q)/2) Object ....
 zeros(p,(c-q)/2)];
 end
end
```
elseif mod(c-q,2)==1
    Impulse=[zeros(p,floor((c-q)/2)) ......
             Impulse zeros(p,floor((c-q)/2)+1)];
end

% Collimating the impulse response and the
% object on the input transparencies. Note
% that, the separation between the impulse
% response and object must be bigger than
% max(Wh,Wg)+Wg/2+Wh/2 in order to make the
% output crosscorrelations of the impulse
% response and object are completely separated
% without any overlapping.
Wh=size(Impulse,1); % The width of the impulse
% response in the direction
% of the y1-coordinate.
Wg=size(Object,1); % The width of the object in
% the direction of the
% y1-coordinate.
Y=max(Wh,Wg)+((Wh+Wg)/2); % The separation
% between the centers
% of the impulse
% response and
% object.
dis=10; % This distance in order to make the
% separation between the centers of the
% impulse response and object bigger
% than max(Wh,Wg)+Wg/2+Wh/2.
Impulse_Object=[............
                zeros(cell (max(Wh,Wg)/2+Wh/4+3/4*Wg)+......
                dis,size(Impulse,2));Impulse;......
                zeros(max(Wh,Wg)+dis,size(Impulse,2));....
                Object;zeros(cell (max(Wh,Wg)/2+......
                3/4*Wh+Wg/4)+dis,size(Object,2))];

% Equalizing the dimensions of the input plane
% P1 by padding it with zeros.
[R C]=size(Impulse_Object);
if R>C
    if mod(R-C,2)==0
        Impulse_Object=[......
                        zeros(size(Impulse_Object,1),.....
                        (R-C)/2) Impulse_Object ....
                        zeros(size(Impulse_Object,1),.....
                        (R-C)/2)];
    elseif mod(R-C,2)==1
        Impulse_Object=[......
                        zeros(size(Impulse_Object,1),.....
                        floor((R-C)/2)) Impulse_Object ....
```
zeros(size(Impulse_Object,1),
    floor((R-C)/2)+1));
end

elseif R<C
    if mod(C-R,2)==0
        Impulse_Object=[zeros((C-R)/2,.....
                        size(Impulse_Object,2));.....
        Impulse_Object;zeros((C-R)/2,.....
        size(Impulse_Object,2))];
    elseif mod(C-R,2)==1
        Impulse_Object=[.....
                        zeros(floor((C-R)/2),.....
                        size(Impulse_Object,2));.....
        Impulse_Object;.....
        zeros(floor((C-R)/2)+1,.....
        size(Impulse_Object,2))];
    end
end

U1=Impulse_Object; % The transmitted field from
% the input plane P1.

[M N]=size(U1);

L=10; % The physical side length of the array
% which holds the input plane P1 in
% meters (m).

dx1_Input=L/N; % The sample spacing in the
% input plane array in the
% direction of the spatial space
% coordinate x1 in meters (m).

dy1_Input=L/M; % The sample spacing in the
% input plane array in the
% direction of the spatial space
% coordinate y1 in meters (m).

x1_Axis_Input=−floor(N/2)*dx1_Input:.....
    dx1_Input:ceil(N/2)*dx1_Input:.....
    dx1_Input; % Sampling the input plane P1 in
    % the direction of the spatial
    % space coordinate x1.

y1_Axis_Input=−floor(M/2)*dy1_Input:.....
    dy1_Input:ceil(M/2)*dy1_Input:.....
    dy1_Input;  % Sampling the input plane P1 in
    % the direction of the spatial
    % space coordinate y1.

U2=fftshift(fft2(fftshift(.....
    U1))); % The Fourier transform of the
% transmitted field in the back
% focal plane of the lens L2.
I=(abs(U2)).^2; % The intensity of the Fourier
% transformed field in the
% plane P2.

lambda=550e-9; % The wavelength in meters (m).
f=0.055; % The focal length in meters (m).
dx2=(lambda*f)/(N*dx1_Input); % The sample spacing in the
% plane P2 in the direction
% of the spatial space
% coordinate x2 in meters
% (m).
dy2=(lambda*f)/(M*dy1_Input); % The sample spacing in the
% plane P2 in the direction
% of the spatial space
% coordinate y2 in meters
% (m).

x2_Axis=floor(N/2)*dx2:dx2:ceil(N/2)*dx2-% Sampling the plane P2 in the
% direction of the spatial space
% coordinate x2.
y2_Axis=floor(M/2)*dy2:dy2:ceil(M/2)*dy2-% Sampling the plane P2 in the
% direction of the spatial space
% coordinate y2.

U3=ifftshift(ifft2(ifftshift(I))); % The crosscorrelated field in the
% back focal plane of the lens L4.

dx3=(lambda*f)/(N*dx2); % The sample spacing in
% the plane P3 in the direction of the
% spatial space
% coordinate x3 in meters (m).
dy3=(lambda*f)/(M*dy2); % The sample spacing in
% the plane P3 in the direction of the
% spatial space
% coordinate y3 in meters (m).

x3_Axis=floor(N/2)*dx3:dx3:ceil(N/2)*dx3-% Sampling the plane P3 in the
% direction of the spatial space
% coordinate x3.
y3_Axis=floor(M/2)*dy3:dy3:ceil(M/2)*dy3-%
dy3; % Sampling the plane P3 in the
direction of the spatial space
coordinate y3.

% Synthesizing a desired filtering mask then
% filtering the crosscorrelated field in the
% plane P3.
Cen=floor(M/2)+1; % The center of the filtering
% mask.
Cen1=Cen−(Y+dis); % The center of the desired
% crosscorrelated field.
Whl=q; % The width of the impulse response in
% the direction of the x1-coordinate.
Wg1=c; % The width of the object in the
% direction of the x1-coordinate.

Mask=zeros(M,N);
for P=Cen1−floor((Wh+Wg)/2):Cen1+......
    ceil((Wh+Wg)/2)
    for Q=Cen−floor((Wh1+Wg1)/2):Cen+......
        ceil((Wh1+Wg1)/2)
            Mask(P,Q)=1;
        end
    end
end

Cross_Field=Mask.*U3; % The filtered
% crosscorrelated field
% in the plane P3.

% For simplicity, instead of processing the
% entire image of the filtered crosscorrelated
% field, we select only the crosscorrelated
% field of interest.
P=Cen1−floor((Wh+Wg)/2):Cen1+ceil((Wh+Wg)/2);
Q=Cen−floor((Wh1+Wg1)/2):Cen+ceil((Wh1+Wg1)/2);
Desired_Cross_Field=U3(P,Q);

Max_Desired_Cross_Fields(n,m)=.....
    max(max(Desired_Cross_Field));

% Testing the process of object recognition.
% Note that, the objects here are known for us in
% order to use them for testing the object recognition
% process as well as setting up a recognition
% threshold.

% The elements in each of the following vectors
% represent the maximum values of the desired
% crosscorrelated fields between the objects and their
% corresponding impulse. Note that, each impulse and
% its corresponding objects have a same face so the
% vectors will include the biggest crosscorrelations.
V1=(Max_Desired_Cross_Fields(1:L1,1)).';
V2=(Max_Desired_Cross_Fields(L1+1:L2,2)).';
V3=(Max_Desired_Cross_Fields(L2+1:L3,3)).';

% The calculation of the mean and the standard
% deviation for each vector of the biggest
% crosscorrelations and stacking them in a vector for
% the means and another for the standard deviations.
% This is done for setting up a threshold for object
% recognition.
V_Mean=[mean(V1);mean(V2);mean(V3)];
V_STD=[std(V1);std(V2);std(V3)];

Failures_Vector=zeros(1,....
Total_No_of_Objects); % A vector for counting the
% number of failures in the
% object recognition process.

for w=1:Total_No_of_Objects
% The object recognition process.
    for ii=1:Total_No_of_Impulses
        if max(Max_Desired_Cross_Fields(w,:))==....
            Max_Desired_Cross_Fields(w,ii) && ....
            max(Max_Desired_Cross_Fields(w,:))>=...
            (V_Mean(ii,1)-V_STD(ii,1)) && ....
            max(Max_Desired_Cross_Fields(w,:))<=...
            (V_Mean(ii,1)+V_STD(ii,1))
            z={'Mr. Mansour Alshammari'; ....
                'Mr. Methkir Alharthee'; ....
                'Mr. Mohammed Hanafy'};
            Recognized_As=char(z(ii,1));
            break
        else
            Recognized_As='Unknown Object';
        end
    end
end

% Defining the object.
if w<=L1
    name='Mr. Mansour Alshammari';
elseif L1<w && w<=L2
name='Mr. Methkir Alharthee';
else name='Mr. Mohammed Hanafy';
end

Str1=strcmp(Recognized_As,...
'Mr. Mansour Alshammari!');
Str2=strcmp(Recognized_As,'Mr. Methkir Alharthee');
Str3=strcmp(Recognized_As,'Mr. Mohammed Hanafy');
F='Success';
if w<=L1 && Str1==0;
    F='Failure';
    Failures_Vector(1,w)=1;
elseif w>L1 && w<=L2 && Str2==0;
    F='Failure';
    Failures_Vector(1,w)=1;
elseif w>L2 && w<=L3 && Str3==0;
    F='Failure';
    Failures_Vector(1,w)=1;
end
end

Total_Number_of_Failures=.....
    sum(Failures_Vector); % The total number of
    % failures in the object
    % recognition process.

Recognition_Error_Rate=(Total_Number_of_Failures/.....
    Total_No_of_Objects)*100; % The error rate
    % of recognition.

% Testing the process of object detection.
% Note that, the objects here are known for us in order
% to use them for testing the object detection process
% as well as setting up a detection threshold.

% The calculation of the mean and standard deviation
% for the crosscorrelations between the objects and
% each impulse response. This is done for setting up a
% threshold for object detection.
Vectorization=reshape(Max_Desired_Cross_Fields,1,.....
    size(Max_Desired_Cross_Fields,1)*.....
    size(Max_Desired_Cross_Fields,2));
Mean=mean(Vectorization);
STD=std(Vectorization);

Failures_Vector1=zeros(1,.....
    Total_No_of_Objects); % A vector for counting the
    % number of failures in the
    % object detection process.
for w=1:Total_No_of_Objects
    % The object detection process.
    if max(Max_Desired_Cross_Fields(w,:))>=....
        (Mean−STD) && ..... 
        max(Max_Desired_Cross_Fields(w,:))<=....
        (Mean+STD)
        Detected_As='a face';
    else
        Detected_As='not a face';
    end
    b='a face'; % The object originally is a face.
    Str=strcmp(Detected_As,'a face');
    F='Success';
    if w<=L3 && Str==0;
        F='Failure';
        Failures_Vector1(1,w)=1;
    end

    Total_Number_of_Failures1=....
    sum(Failures_Vector1); % The total number of
    % failures in the object
    % detection process.
    Detection_Error_Rate=(Total_Number_of_Failures1/.....
    Total_No_of_Objects)*100; % The error rate
    % of detection.
    Recognition_Combinations((Im3−L2)+i,:)=.....
    [Impulses Recognition_Error_Rate];

    Detection_Combinations((Im3−L2)+i,:)=.....
    [Impulses Detection_Error_Rate];

    end
    i=i+(Im3−L2);
end

Im1

save('Impulses Combinations for Recognition',....
    'Recognition_Combinations')
save('Impulses Combinations for Detection',....
    'Detection_Combinations')
% Finding the optimal combination of impulses which obtain the % lowest error rate of recognition.
Optimal_Combination_Recognition=0; % The optimal combination of % impulses for recognition.
Optimal_Error_Rate_Recognition=0; % The lowest error rate of % recognition that is produced % when the optimal combination % of impulses for recognition % is used.
for j=1:size(Recognition_Combinations,1)
    if Recognition_Combinations(j,Total_No_of_Impulses+1)==...
        min(Recognition_Combinations(:,Total_No_of_Impulses+1))
        Optimal_Combination_Recognition=.......% Recognition_Combinations(j,1:Total_No_of_Impulses);
        Optimal_Error_Rate_Recognition=.....% Recognition_Combinations(j,Total_No_of_Impulses+1);
    end
end

% Finding the optimal combination of impulses which obtain the % lowest error rate of detection.
Optimal_Combination_Detection=0; % The optimal combination of % impulses for detection.
Optimal_Error_Rate_Detection=0; % The lowest error rate of % detection that is produced when % the optimal combination % of impulses for detection is used.
for j=1:size(Detection_Combinations,1)
    if Detection_Combinations(j,Total_No_of_Impulses+1)==....
        min(Detection_Combinations(:,Total_No_of_Impulses+1))
        Optimal_Combination_Detection=.......
        Detection_Combinations(j,1:Total_No_of_Impulses);
        Optimal_Error_Rate_Detection=.....
        Detection_Combinations(j,Total_No_of_Impulses+1);
    end
end
Appendix K. Authorizations for Using People Photos

Authorizations for Using People Photos

Figure K.1, Figure K.2 and Figure K.3 show respectively the authorizations for using the photos of Mr. Mansour Thuwaini Al-Shammari, Mr. Mathkar Alawi Alharthi and Mr. Mohamed Elsayed Hanafy.

I am Mansour Thuwaini Al-Shammari. I give Abdulaziz Abdullah Alorf full authorization to use the images of my face in his researches.

Signature,

Figure K.1: The authorization for using the photos of Mr. Mansour Thuwaini Al-Shammari.
I am Mathkar Alawi Alharthi. I give Abdulaziz Abdullah Alorf full authorization to use the images of my face in his researches.

Signature,

Obscured

Figure K.2: The authorization for using the photos of Mr. Mathkar Alawi Alharthi.

I am Mr. Mohamed Elsayed Hanafy. I give Abdulaziz Abdullah Alorf full authorization to use the images of my face in his researches.

Signature,

Obscured

Figure K.3: The authorization for using the photos of Mr. Mohamed Elsayed Hanafy.