



**BIOMETRIC FACE RECOGNITION USING  
MULTILINEAR PROJECTION AND  
ARTIFICIAL INTELLIGENCE**

By

*Abeer A. Mohamad AL-Shiha*

A THESIS  
SUBMITTED TO THE FACULTY OF  
SCIENCE, AGRICULTURE AND ENGINEERING  
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

**DOCTOR OF PHILOSOPHY**

SCHOOL OF ELECTRICAL, ELECTRONIC & COMPUTER ENGINEERING  
NEWCASTLE UNIVERSITY  
UNITED KINGDOM

---

July 2013



**“He gives wisdom to whom He wills, and whoever has been given  
wisdom has certainly been given much good”**

*Sorah AL-Baqarah - Ayaa 269”*

NEWCASTLE UNIVERSITY  
SCHOOL OF ELECTRICAL, ELECTRONIC &  
COMPUTER ENGINEERING

I, *Abeer A. Mohamad AL-Shiha*, confirm that this thesis and work presented in it are my own achievement.

I have read and understand the penalties associated with plagiarism.

Signature:

Date: July 2013

## SUPERVISOR'S CERTIFICATE

This is to certify that the entitled thesis "Biometric Face Recognition Using Multilinear Projection and Artificial Intelligence " has been prepared under my supervision at the school of Electrical, Electronic and Computer Engineering/ Newcastle University for the degree of PhD in Computer Engineering- Artificial Intelligence.

Signature:

Supervisor: Professor Satnam Dlay

Date: July 2013

Signature:

Student: Abeer A. Mohamad AL-Shiha

Date: July 2013

# ABSTRACT

---

Numerous problems of automatic facial recognition in the linear and multilinear subspace learning have been addressed; nevertheless, many difficulties remain. This work focuses on two key problems for automatic facial recognition and feature extraction: object representation and high dimensionality.

To address these problems, a bidirectional two-dimensional neighborhood preserving projection (B2DNPP) approach for human facial recognition has been developed. Compared with 2DNPP, the proposed method operates on 2-D facial images and performs reductions on the directions of both rows and columns of images. Furthermore, it has the ability to reveal variations between these directions. To further improve the performance of the B2DNPP method, a new B2DNPP based on the curvelet decomposition of human facial images is introduced. The curvelet multi-resolution tool enhances the edges representation and other singularities along curves, and thus improves directional features. In this method, an extreme learning machine (ELM) classifier is used which significantly improves classification rate. The proposed C-B2DNPP method decreases error rate from 5.9% to 3.5%, from 3.7% to 2.0% and from 19.7% to 14.2% using ORL, AR, and FERET databases compared with 2DNPP. Therefore, it achieves decreases in error rate more than 40%, 45%, and 27% respectively with the ORL, AR, and FERET databases.

Facial images have particular natural structures in the form of two-, three-, or even higher-order tensors. Therefore, a novel method of supervised and unsupervised multilinear neighborhood preserving projection (MNPP) is proposed for face recognition. This allows the natural representation of multidimensional images 2-D, 3-D or higher-order tensors and extracts useful information directly from tensorial data rather than from matrices or vectors. As opposed to a B2DNPP which derives only two subspaces, in the MNPP method multiple interrelated subspaces are obtained over different tensor directions, so that the subspaces are learned iteratively by unfolding the tensor along the different directions. The performance of the MNPP has performed in terms of the two modes of facial recognition biometrics systems of identification and verification. The proposed supervised MNPP method achieved decrease over 50.8%, 75.6%, and 44.6% in error rate using ORL, AR, and FERET databases respectively, compared with 2DNPP. Therefore, the results demonstrate that the MNPP approach obtains the best overall performance in various learning scenarios.

## DEDICATION

*To my beloved family*

*Thanks for your*

*unconditional love,*

*constant support,*

*and encouragement.*

*I love you all dearly.*



## **ACKNOWLEDGEMENT**

---

In the name of Allah, the Beneficent, the Merciful. Praise and Gratitude be to Allah (SWT) for giving me strength and guidance, so that this dissertation can be finished accordingly.

This thesis is a collection of not only hard work, perseverance and continuous effort over the past four years, but also encouragement, cooperation and support from many people who deserve to be acknowledged.

First and foremost, I would like to express my sincere gratitude to my supervisors Professor Satnam Dlay and Dr. Wai Lok Woo for their guidance, nurturing, encouragement, and support at every stage of my graduate study. Their attention to detail, quest for excellence, love of perfection and desire for hardwork has inspired me to give my very best during the time of my PhD research programme. My deepest appreciation for both of them for training me to critically analyse scientific issues and nurturing me in becoming an independent researcher. I am very fortunate to have had them as my supervisors. I would like to express my utmost gratitude to Professor Christian Kray who has been always supportive in many respects over all these years. Also, I would like to thank Mrs. Gillian Webber for her constant encouragement, support and her warm heart during the course of my programme. I wish you all the best and further success and achievements in your life.

It is not enough to just say thank you to those who have made many sacrifices to help me: my parents, husband, brothers, sisters and my children. The only thing I can say to you is that your dream to see me become a scientist will soon be real. I love you, not only from the bottom of my heart, but from the core of my soul.

Grateful thanks to my beloved children, Mustafa and Tuka, who have lived this journey with me, in times of happiness and hardship p. Thank you very much for being the motivational factor in my life, supporting me in every possible way and giving me hope after hope to achieve this work successfully, I am really proud of you. Please forgive me for my shortcomings, which were many, through this journey.

To all my team and my friends: thank you very much for what you have done for me. I thank you all for the companionship that made this journey much easier. In fact, I do not need to list your names because I am sure that you know who you are.

Last, but not least, I dedicate this research to my home country of Iraq. I also thank the Iraqi Cultural Attaché, London, for supporting me during my study abroad.

---

# LIST OF CONTENTS

---

<b>CHAPTER 1: INTRODUCTION</b> .....	<b>1</b>
1.1 AN OVERVIEW OF THE BIOMETRICS SYSTEM .....	2
1.1.1 <i>What is Biometrics?</i> .....	2
1.1.2 <i>Authentication and Biometrics</i> .....	3
1.1.3 <i>Characteristic Needed by Biometrics Systems</i> .....	4
1.1.4 <i>Types of Biometrics Systems</i> .....	5
1.2 BIOMETRIC FACIAL RECOGNITION .....	8
1.2.1 <i>Approaches to Face Recognition</i> .....	10
1.2.2 <i>Challenges</i> .....	11
1.3 MOTIVATION .....	12
1.3.1 <i>Face Image Representation</i> .....	12
1.3.2 <i>Linear subspace learning</i> .....	13
1.3.3 <i>Multilinear subspace learning</i> .....	14
1.4 CONTRIBUTION.....	15
1.5 ORGANIZATION OF THE THESIS .....	17
<b>CHAPTER 2: LINEAR SUBSPACE LEARNING</b> .....	<b>19</b>
2.1 INTRODUCTION.....	19
2.1.1 <i>Feature Extraction</i> .....	22
2.1.2 <i>Feature Selection Process</i> .....	22
2.2 LINEAR SUBSPACE LEARNING APPROACH.....	23
2.2.1 <i>LSL problem definition</i> .....	24
2.2.2 <i>Motivation for the use linear-based algorithms</i> .....	24
2.3 RELEVANT PREVIOUS WORK ON LSL .....	26
2.3.1 <i>Principle Component Analysis (PCA)</i> .....	27
2.3.2 <i>Linear Discriminant Analysis (LDA)</i> .....	29
2.3.3 <i>Locality Preserving Projection (LPP)</i> .....	31
2.3.4 <i>Neighborhood Preserving Projections (NPP)</i> .....	33
2.4 TWO-DIMENSIONAL LINEAR SUBSPACE LEARNING APPROACHES.....	38
2.4.1 <i>Two-Dimensional LSL Based Methods</i> .....	38
2.4.2 <i>Bidirectional Two-Dimensional Compression Methods</i> .....	39
2.5 SUMMARY .....	41

<b>CHAPTER 3: MULTILINEAR SUBSPACE LEARNING, AND FACIAL</b>	
<b>DATABASE .....</b>	<b>42</b>
3.1 INTRODUCTION.....	42
3.2 BASIS OF THE MULTILINEAR APPROACH .....	43
3.2.1 <i>Notations</i> .....	44
3.2.2 <i>Multilinear algebra</i> .....	45
3.3 MULTILINEAR PROJECTION .....	50
3.4 MULTILINEAR SUBSPACE LEARNING APPROACH .....	51
3.4.1 <i>MSL problem definition</i> .....	51
3.4.2 <i>General multilinear approach</i> .....	52
3.5 RELEVANT MULTILINEAR ALGORITHMS IN PREVIOUS RESEARCHES .....	53
3.5.1 <i>Multilinear principal component analysis (MPCA)</i> .....	54
3.5.2 <i>Discriminant analysis with tensor representation (DATER)</i> .....	56
3.5.3 <i>Multilinear tensor supervised neighborhood embedding (ND-TSNE)</i> .....	57
3.6 FACIAL DATABASES .....	57
3.6.1 <i>The ORL database</i> .....	58
3.6.2 <i>The AR database</i> .....	58
3.6.3 <i>The FERET database</i> .....	59
3.6.4 <i>Dataset configuration</i> .....	60
3.7 PRE-PROCESSING OF FACIAL IMAGES .....	60
3.8 PERFORMANCE EVALUATION OF FACIAL RECOGNITION.....	61
3.9 SUMMARY .....	62
<b>CHAPTER 4: FACE RECOGNITION APPROACH BASED ON</b>	
<b>BIDIRECTIONAL TWO-DIMENSIONAL NEIGHBORHOOD</b>	
<b>PRESERVING PROJECTION AND CURVELET TRANSFORM.....</b>	<b>64</b>
4.1 INTRODUCTION.....	65
4.2 CURVELET MULTI-RESOLUTION TRANSFORM .....	69
4.2.1 <i>Theory of curvelet transform</i> .....	69
4.2.2 <i>Implementation of the curvelet transform</i> .....	72
4.3 BIDIRECTIONAL TWO-DIMENSIONAL NEIGHBORHOOD PRESERVING	
PROJECTION BASED ON CURVELET TRANSFORM (C-B2DNPP) .....	74
4.3.1 <i>The column-direction 2DNPP</i> .....	75

4.3.2	<i>Bidirectional two-dimensional neighborhood preserving projection (B2DNPP)</i> .....	76
4.3.2.1	Formulation of the problem of dimensionality reduction .....	77
4.3.2.2	The B2DNPP projection learning .....	77
4.4	EXTREME LEARNING MACHINE (ELM) CLASSIFIER.....	80
4.5	THE PROPOSED C-B2DNPP METHOD FOR FACE RECOGNITION.....	83
4.5.1	<i>C-B2DNPP training phase</i> .....	83
4.5.2	<i>C-B2DNPP recognition phase</i> .....	86
4.6	EXPERIMENT RESULTS .....	88
4.6.1	<i>Proposed algorithm and setting</i> .....	88
4.6.2	<i>Experiment using ORL synthetic data</i> .....	94
4.6.3	<i>Experiment using AR synthetic data</i> .....	96
4.6.4	<i>Experiment using FERET synthetic data</i> .....	96
4.6.5	<i>Dimensionality selection</i> .....	98
4.6.6	<i>Discussion</i> .....	101
4.7	SUMMARY .....	103

**CHAPTER 5: MULTILINEAR NEIGHBORHOOD PRESERVING**

**PROJECTION FOR FACE RECOGNITION ..... 105**

5.1	INTRODUCTION.....	106
5.2	MULTI-DIMENSIONAL NEIGHBORHOOD PRESERVING PROJECTION APPROACH (MNPP).....	108
5.2.1	<i>Multilinear dimensionality reduction problem formulation</i> .....	110
5.2.2	<i>Neighborhood weighting</i> .....	110
5.2.3	<i>The derivation of MNPP solution and projection learning</i> .....	111
5.2.4	<i>Supervised MNPP approach</i> .....	115
5.3	DESIGN OF THE MNPP APPROACH.....	117
5.3.1	<i>Initial projection</i> .....	117
5.3.2	<i>Iterative algorithm</i> .....	117
5.3.3	<i>Termination</i> .....	119
5.3.4	<i>Convergence in the MNPP method</i> .....	119
5.3.5	<i>Projection order</i> .....	120
5.4	MNPP-BASED TENSOR OBJECT RECOGNITION .....	120
5.5	EVALUATION OF THE PERFORMANCE OF MNPP.....	121
5.6	EXPERIMENTS AND ANALYSIS.....	123

<i>List of Contents</i> .....	
5.6.1 <i>Databases and experimental design</i> .....	124
5.6.1.1 Experiment on the ORL Database.....	126
5.6.1.2 Experiment on AR Databasea .....	127
5.6.1.2.1 AR Database with varying facial expressions (AR1) .....	127
5.6.1.2.2 AR database with varying lighting conditions (AR2).....	128
5.6.1.2.3 AR database under varying occlusions (AR3) .....	129
5.6.1.3 Experiment on the FERET Database .....	130
5.6.2 <i>Experiments on Identification system</i> .....	132
5.6.3 <i>Experiments on the verification system</i> .....	138
5.6.4 <i>Dimensionality selection</i> .....	153
5.6.5 <i>Discussion</i> .....	155
5.7 SUMMARY .....	158
<b>CHAPTER 6: CONCLUSION AND FUTURE WORK</b> .....	<b>159</b>
6.1 SUMMARY AND MAJOR CONTRIBUTIONS OF THE RESEARCH.....	159
6.2 FUTURE RESEARCH .....	162
<b>REFERENCES</b> .....	<b>166</b>

# LIST OF ACRONYMS

<b>Acronyms</b>	<b>Description</b>
1-D	One Dimensional Vector
2-D	Two Dimensional Matrix
2DNPP	Two-Dimensional Neighborhood Preserving Projection
3-D	Three-Dimensional
AI	Artificial Intelligence
ALS	Alternating Least Square
ANN	Artificial Neural Networks
ATM	Automated Teller Machine
B2DNPP	Bidirectional Two-Dimensional Neighborhood Preserving Projection
BP	Back-Propagation
C-B2DNPP	Bidirectional Two-Dimensional Neighborhood Preserving Projection Based on Curvelet Transform
CMC	Cumulative Match Characteristic
CRR	Correct Recognition Rate
DATER	Discriminant Analysis with Tensor Representation
DET	Detection Error Trade-Off
EER	Equal Error Rates
ELM	Extreme Learning Machine
ERR	Error Rate
FA	False Acceptance

<b>Acronyms</b>	<b>Description</b>
FAR	False Accept Rate
FDCT	Fast Digital Curvelet Transform
FFT	Fast Fourier Transform
FR	False Rejection
FRR	False Reject Rate
GAR	Genuine Acceptance Rate
HCI	Human-Computer Interaction
ID	Identification Card
Isomap	Isometric Feature Mapping
<i>K-Nns</i>	<i>K</i> Nearest Neighbors
LDA	Linear Discriminate Analysis
LE	Laplacian Eigenmap
LLE	Locally Linear Embedding
LPP	Local Preserving Projection
LSL	Linear Subspace Learning
LS-SVM	Least Square Support Vector Machine
MNPP	Multilinear Neighborhood Preserving Projection Approach
MPCA	Multilinear Principal Component Analysis
MSL	Multilinear Subspace Learning
ND-TSNE	Multilinear Supervised Neighborhood Embedding
NPP	Neighborhood Preserving Projection
ONPP	Orthogonal Neighborhod Preserving Projection

PCA	Principal Component Analysis
Pcs	Principal Components
PIN	Personal Identification Number
RMSE	Root Mean Square Error
ROC	Receiver Operator Characteristic
SLFN	A Single Hidden Layer Feedforward Neural Network
SSS	Small Sample Space
TA	True Acceptance
USFFT	Unequally Spaced Fast Fourier Transform

# LIST OF NOTATIONS

Symbol	Description
$n$	The mode index of a tensor object
$N$	The order of a tensor object, the number of indices/modes
$I_1, I_2, \dots, I_n$	The dimensionality of the $n$ -mode
$P, P_1, P_2, \dots, P_n$	The $n$ -mode dimensionality in the projected space
$\text{vec}(\mathcal{A})$	The vectorized representation of the tensor $\mathcal{A}$
$\  \cdot \ _F$	The Frobenius norm
$\mathcal{X}_m$	The $m$ th input tensor sample
$\mathbf{X}_m$	The $m$ th input matrix sample
$\mathbf{U}^{(n)}$	The $n$ th projection matrix
$\mathcal{Y}_m$	The projection of $\mathcal{X}_m$ on $\{\mathbf{U}^{(n)}\}$
$C$	The number of classes in the facial dataset
$c$	The index of the class
$c_m$	The class label for the $m$ th training image sample
$M_c$	The number of training images in class $c$
$\bar{\mathbf{x}}$	The mean of the samples
$\bar{\mathbf{x}}_{c_m}$	The mean vector of samples in class $c$
$\mathbf{S}_B$	The between-class scatter of samples
$\mathbf{S}_W$	The within-class scatter of samples
$\lambda$	The eigenvalue solution to the optimization problem

<b>Symbol</b>	<b>Description</b>
$\varepsilon$	A small constant number
<b>G</b>	The local Gramian matrix
$\mathcal{G}$	A nearest neighbor graph
<b>W</b>	The optimal weight matrix of $\mathcal{G}$
$w_{ij}$	The optimal weight of the edge between the tensor $i$ and tensor $j$ .
$k$	The number of the nearest neighbor images
$M$	The number of training samples for each class (subject)
$T$	The maximum number of iterations in multilinear solutions

## LIST OF PUBLICATIONS

---

- 1- Al-Shiha, A.M.; Dlay, S.; Woo, W. L., "Design an Artificial Mirror system as an interaction method for the life logging system," *International Conference on User Science and Engineering (i-USEr)*, PP128- 132, 2010.
- 2- Al-Shiha, A.M.; Woo, W. L.; Dlay, S., "Multilinear Neighborhood Preserving Projection for Face Recognition", *Submitted to Pattern Recognition Elsevier*, under review, 2012.
- 3- Al-Shiha, A.M.; Dlay, S.; Woo, W. L., "Face Recognition Approach Based on Bidirectional Two-Dimensional Neighborhood Preserving Projection and Curvelet Transform", *Submitted to IEEE Transaction on image processing*, under review, 2013.

# LIST OF TABLES

---

Table 1.1: Categorization of Different Algorithms.....	15
Table 2.1: Algorithm of the NPP Approach.....	36
Table 2.2: Summary of Common LSL Methods for Linear Feature Extraction.....	38
Table 3.1: Overview of existing linear and multilinear subspace learning approaches. The pink shaded empty boxes indicate a lack of relevant studies so far. ....	54
Table 4.1: Variation of curvelet sub-band 1 in different databases .....	90
Table 4.2: Variation of curvelet sub-band 2 in different databases .....	90
Table 4.3: Variation of curvelet sub-band 3 in different databases .....	91
Table 4.4: Variation of curvelet sub-band 4 in different databases .....	91
Table 4.5: Error rate (%) in each sub-band for the ORL database.....	92
Table 4.6: Error rate (%) in each sub-band for the AR database. ....	92
Table 4.7: Error rate (%) in each sub-band for the FERET database. ....	92
Table 4.8: Comparison of error rates (%) using the ORL database. ....	94
Table 4.9: Comparison of error rates (%) using the AR database. ....	96
Table 4.10: Comparison of error rates (%) using the FERET database.....	97
Table 5.1: Categorization of Different Methods .....	108
Table 5.2: Performance comparisons of the error rate (%) using the ORL database....	132
Table 5.3: Performance comparisons of the error rate (%) using the AR database under varying facial expressions.....	133

Table 5.4: Performance comparisons of the error rate (%) using the AR colour database under varying facial expressions.....	134
Table 5.5: Performance comparisons of the error rate (%) using the AR database with varying lighting conditions .....	135
Table 5.6: Performance comparisons of the error rate (%) using the AR colour database with varying lighting conditions .....	135
Table 5.7: Performance comparisons of the error rate (%) using the AR database with varying occlusion.....	136
Table 5.8: Performance comparisons of the error rate (%) using the AR colour database with varying occlusion .....	136
Table 5.9: Performance comparisons of the error rate (%) using the FERET gray database.....	137
Table 5.10: Performance comparisons of the error rate (%) using the FERET colour database .....	137
Table 5.11: EER and GAR of the NPP, 2DNPP and the proposed MNPP method with the ORL database.....	139
Table 5.12: Values of EER and GAR for the NPP, 2DNPP and the proposed MNPP method with AR grey database and varying facial expressions ..	141
Table 5.13: Values of EER and GAR for the NPP, 2DNPP and proposed MNPP method with the AR grey database and varying lighting conditions .....	142
Table 5.14: Values of EER and GAR for the NPP, 2DNPP and proposed MNPP method with the AR grey database and varying occlusion.....	144
Table 5.15: Values of EER and GAR for the NPP, 2DNPP and the proposed MNPP method with the FERET grey database .....	145
Table 5.16: Values of EER and GAR for the MPCA, DATER and proposed MNPP method on AR colour database and varying facial expressions ..	147

Table 5.17: Values of EER and GAR for the MPCA, DATER and proposed MNPP method with the AR colour database and varying lighting conditions.....	149
Table 5.18: Values of EER and GAR for the MPCA, DATER and proposed MNPP method with the AR colour database and varying occlusion.....	150
Table 5.19: Values of EER and GAR for the MPCA, DATER and proposed MNPP method with the FERET colour database. ....	152

# LIST OF FIGURES

---

Figure 1.1: Types of Biometrics.....	3
Figure 1.2: Enrolment, Verification, Identification and Matching Modules of a General Biometrics System. Here, $U$ represent the biometric trait obtained during enrolment, $Q$ is the query biometric trait obtained during the recognition stages, and $T_U$ and $T_Q$ are the biometric template and the query feature sets respectively. ....	7
Figure 1.3: Identification (ID) card produced by the Immigration Office in the United Kingdom .....	9
Figure 2.1: The Block diagram of Eigenface Recognition Technique.....	29
Figure 2.2: Between-class variance and within-class variance before and after using LDA.....	29
Figure 2.3: The Block Diagram of the LDA Linear Subspace Learning Approach. ....	31
Figure 2.4: Outlines of the NPP approach .....	37
Figure 3.1: Real-world images in their natural tensor representation: (a) a colour- level image, (b) a grey-level image, and (c) 3-channels (red, green, and blue) of the colour-level image. ....	45
Figure 3.2: Example of mode- $n$ vectors: (a) a tensor $\mathcal{X} \in R^{6 \times 5 \times 3}$ , (b) the mode-1 slices, (c) the mode-2 slices, and (d) the mode-3 slices.....	46
Figure 3.3: Example of unfolding the third-order tensor to the three mode- $n$ matrices: (a) a tensor $\mathcal{X} \in R^{6 \times 5 \times 3}$ , (b) mode-1 matrix (columns), (c) mode-2 matrix (rows), and (d) mode-3 matrix (colours).....	47
Figure 3.4: Projection of a 3-order tensor in all modes.....	50
Figure 3.5: Projection of a 3-order tensor in mode-1 .....	51

Figure 3.6: Typical multilinear subspace learning algorithm pseudo-code. ....	53
Figure 3.7: Facial samples from four subjects within the ORL database .....	58
Figure 3.8: AR colour dataset sample with variations in facial expressions, lighting conditions and occlusions.....	59
Figure 3.9: Twenty four facial samples of one subject from the FERET subset. ....	60
Figure 3.10: The pre-processing procedure for face images.....	61
Figure 4.1: Two levels wavelet decomposition of a two-dimensional facial image.....	67
Figure 4.2: Curvelet space-frequency tiling, (a): discrete domain, (b): continuous domain. The gray shaded areas represent wedges. ....	70
Figure 4.3: Wrapping a segment around a discrete localization window: (a) the original before wrapping and (b) after wrapping.....	74
Figure 4.4: Block diagram of the proposed B2DNPP method.....	78
Figure 4.5 : The architecture of an ELM classifier .....	81
Figure 4.6: Methodology of the proposed C-B2DNPP method.....	84
Figure 4.7: Example of curvelet coefficients at 4 scales: (a) original image, (b) coarse and fine curvelet coefficients (sub-bands 1-3), and (c) finest coefficient (sub-band4). ....	86
Figure 4.8: Curvelet coefficients at 4 scales and different directions. ....	90
Figure 4.9: Error rate (%) performance of C-B2DNPP and two dimensional methods for ORL database .....	95
Figure 4.10: Error rate (%) performance of C-B2DNPP and bidirectional two- dimensional methods for ORL database .....	95
Figure 4.11: A sample from the AR dataset.....	96

Figure 4.12: Error rate (%) performance of C-B2DNPP and two dimensional methods for FERET database .....	97
Figure 4.13: Error rate (%) performance of C-B2DNPP and bidirectional two-dimensional methods for FERET database.....	98
Figure 4.14: Error rate (%) against number of eigenvectors for ORL database. ....	99
Figure 4.15: Error rate (%) against number of eigenvectors for AR database.....	100
Figure 4.16: Error rate (%) against number of eigenvectors for FERET database. ....	100
Figure 5.1: Block Diagram of MNPP System for Face Recognition.....	109
Figure 5.2: The Pseudo Code for the MNPP Proposed Approach.....	116
Figure 5.3: Illustration of FRR, FAR, GAR and EER parameters at a given threshold.....	122
Figure 5.4: Some facial images from the ORL database .....	126
Figure 5.5: Contents of the ORL outer set. ....	126
Figure 5.6: AR colour dataset sample with variations in facial expressions, lighting conditions and occlusions.....	127
Figure 5.7: Contents of the AR outer set with varying facial expressions.....	128
Figure 5.8: Contents of the AR outer set with different lighting conditions .....	129
Figure 5.9: Contents of the AR outer set with varying occlusions .....	130
Figure 5.10: Contents of the FERET outer set.....	131
Figure 5.11: Facial samples of a subject from the FERET subset .....	131
Figure 5.12: Comparisons of values of TA, FR, and FA the of the NPP, 2DNPP and the proposed MNPP method with the ORL database .....	140

Figure 5.13: Comparisons of the ROC curves for the NPP, 2DNPP and proposed MNPP method with ORL database.....	140
Figure 5.14: Comparisons of TA, FR, and FA values for the NPP, 2DNPP and proposed MNPP method with AR grey database and varying facial expressions.....	141
Figure 5.15: Comparisons of ROC curves for the NPP, 2DNPP and proposed MNPP method with the AR grey database and varying facial expressions.....	142
Figure 5.16: Comparisons of TA, FR, and FA values for the NPP, 2DNPP and proposed MNPP method with the AR grey database and varying lighting conditions .....	143
Figure 5.17: Comparisons of the ROC curves for the NPP, 2DNPP and the proposed MNPP method with the AR grey database and varying lighting conditions .....	143
Figure 5.18: Comparisons of TA, FR, and FA values for the NPP, 2DNPP and the proposed MNPP method with the AR grey database and varying occlusion .....	144
Figure 5.19: Comparisons of ROC curves for the NPP, 2DNPP and the proposed MNPP method with the AR grey database and varying occlusion.....	145
Figure 5.20: Comparisons of TA, FR, and FA values for the NPP, 2DNPP and the proposed MNPP method with the FERET grey database.....	146
Figure 5.21: Comparisons of ROC curves for the NPP, 2DNPP and the proposed MNPP method with the FERET grey database. ....	146
Figure 5.22: Comparisons of TA, FR, and FA values for the MPCA, DATER and proposed MNPP method with the AR colour database and varying facial expressions.....	148

Figure 5.23: Comparisons of ROC curves of the of the MPCA, DATER and the proposed method MNPP on AR colour database under varying facial expressions.....	148
Figure 5.24: Comparisons of TA, FR, and FA values for the MPCA, DATER and proposed MNPP method with the AR colour database and varying lighting conditions .....	149
Figure 5.25: Comparisons of the ROC curves for MPCA, DATER and proposed MNPP method with the AR colour database and varying lighting conditions.....	150
Figure 5.26: Comparisons of TA, FR, and FA values for the MPCA, DATER and proposed MNPP method with AR colour database and varying occlusion .....	151
Figure 5.27: Comparisons of the ROC curves for the MPCA, DATER and proposed MNPP method with the AR colour database and varying occlusion .....	151
Figure 5.28: Comparisons of TA, FR, and FA values for the MPCA, DATER and proposed MNPP method with the FERET colour database.....	152
Figure 5.29: Comparisons of the ROC curves for the MPCA, DATER and proposed MNPP method with the FERET colour database under.....	153
Figure 5.30: Number of eigenvectors selected against detection recognition rates with the colour FERET database. ....	154
Figure 6.1: Parallel model for facial recognition .....	163
Figure 6.2: Parallel multidimensional model for facial recognition .....	164

# Chapter 1:

## Introduction

---

For thousands of years, humans have used visual body characteristics, such as the features of the face to recognize people. This amazing ability has inspired researchers to build automated systems that have the ability to recognize individuals from facial images [1]. Therefore, the observation and interpretation of how people perform facial matching is an essential research topic for artificial intelligence (AI) scientists, and it is likely to be a good starting point for automatic attribute analysis in order to take advantage of the benefits of present technologies. The AI involves the use of technology to assist machines in order to discover suitable solutions to complex problems in a more human-like way. This involves borrowing human skills and applying them as efficient and flexible artificial algorithms in a computer. These days, AI is applied in many aspects of life and it has a wide range of applications such as in medical diagnosis, biometrics, robot control, games, and law enforcement. Face recognition belong to the biometrics, a very active area of research in the computer vision and pattern recognition society [2]. Therefore, this thesis focus on multilinear subspace learning, the problem of learning low dimensional representations using the natural tensorial data, for biometric face recognition as one of the artificial intelligence applications [1, 3].

This chapter reviews biometric face recognition technology which has recently received significant attention. Biometric systems and their applications are introduced with particular reference to biometrics facial recognition problems. Then the key technical challenges in solving such problems are outlined. Next, the motivation for this study is described. Popular subspace learning techniques are discussed with a focus on multilinear subspace learning approaches which avoid the fundamental limitations of traditional linear subspace learning algorithms by directly using the natural tensoral objects. Finally, the contributions of this study are highlighted, and the structure of the thesis is explained.

## 1.1 An Overview of the Biometrics System

Biometrics technology shows increasing promise in network security. In the next few years it will play an increasingly vital role as a secure authentication system for different applications. This section introduces biometrics technologies and describes the general biometrics techniques utilized in authentication systems.

### 1.1.1 What is Biometrics?

With the increasing growth of information technology in the digital age, reliable identity authentication is required in many areas. The key problem to be solved here is how to recognize a person's identity accurately while ensuring the security of information. Therefore, a range of inherent human biometric characteristics are studied, which led to the development of biometric identification technology. Biometrics is a user-friendly authentication technology that uses biological data to identify individual persons. It refers to automated tools that measure and statistically analyse distinctive human characteristics in order to identify people for authentication purposes [3]. Therefore, instead of requiring personal identification cards, keys or passwords, biometrics systems use computer technology to identify individuals depending on their physiological or behavioural properties as shown in Figure 1.1. Bodily traits that can be used for biometric recognition systems include:

- 1- **Fingerprint:** the analysis of a person's fingerprint.
- 2- **Facial recognition:** the analysis of facial characteristics.
- 3- **Voice pattern:** the analysis of the tone, cadence and frequency of a person's voice.
- 4- **Hand geometry:** the analysis of the shape of the hand, its width, curvatures, thickness and the length of fingers in an individual's hand.
- 5- **Iris:** the analysis of the coloured ring that surrounds the pupil in the eye.
- 6- **Signature:** the analysis of the way that the individual signs his name, such as the speed, direction and pressure of writing.
- 7- **Gait:** analysis of an individual's walking style.
- 8- **Keystroke analysis:** analysis of the individual's keystrokes on a keyboard.
- 9- **DNA:** analysis of an individual's genetic code.
- 10- **Palm print:** analysis of information about the texture of the individual's palm.

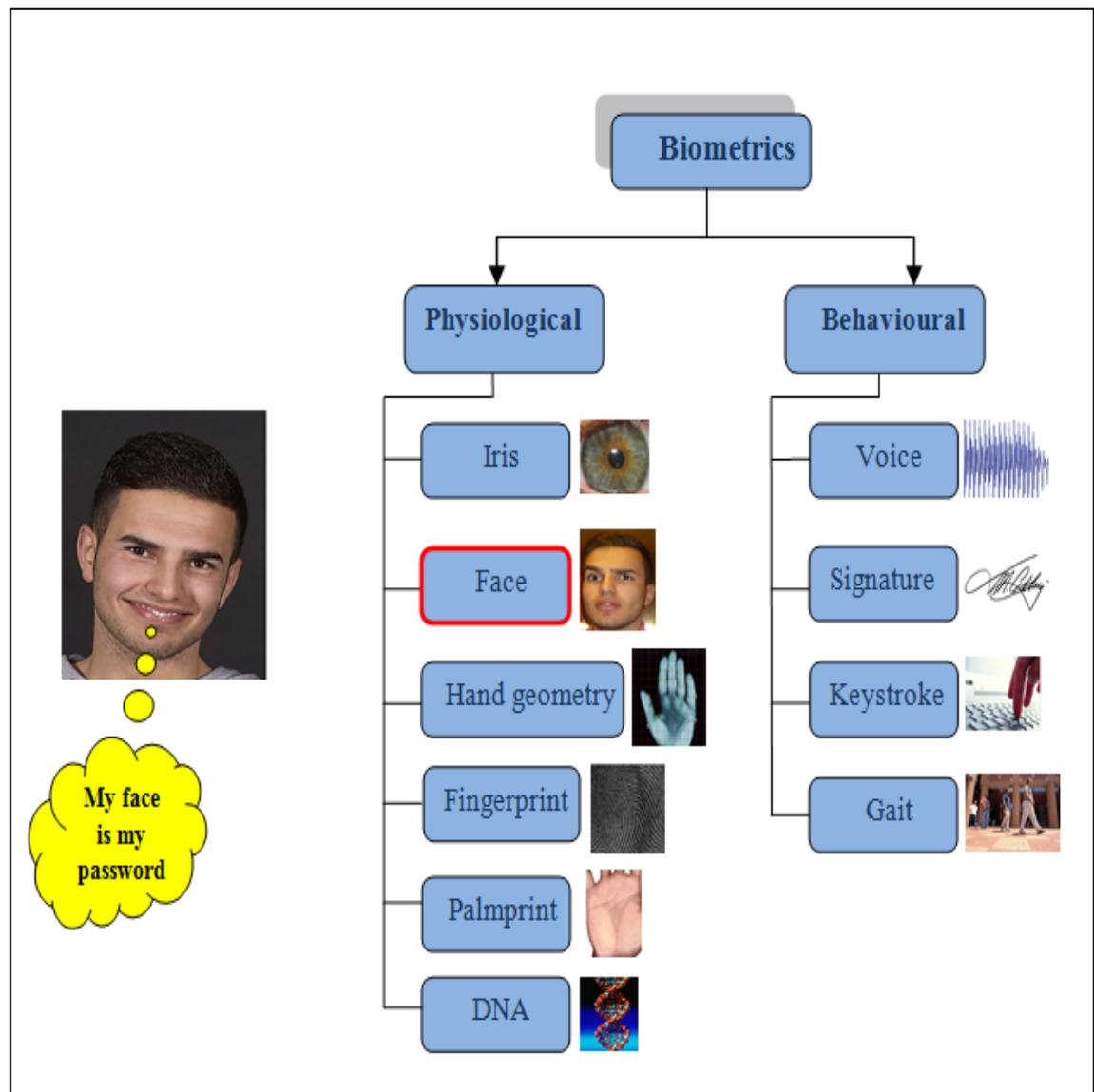


Figure 1.1: Types of Biometrics

### 1.1.2 Authentication and Biometrics

Authentication is a process by which a system confirms the identity of a user who desires to access it, in order to ensure that the given services are accessed only by authorised persons and not any other users. Authentication is necessary to ensure secure access for example, to computer accounts such as the PC login and internet transactions, secure areas in buildings, on-line banking and bank automated teller machine (ATMs). In general, there are three types of authentication approaches, as follows:

1. What you possess such as a key, card or token.
2. What you remember, such as a password and personal identification number (PIN).

3. What you do or who you are, such as biometrics.

A classical system based on handheld tokens such as keys and ID cards and knowledge-based systems such as passwords and PINs have been used extensively to control access. Nevertheless, these authentication methods are not sufficient for reliable identity determination because they can be easily shared, stolen, misplaced, forgotten or forged, as with ID cards. Short and simple passwords can be guessed, whereas better and longer passwords are hard to remember by legitimate users. On the other hand, a machine using biometrics can automatically and electronically recognize a user, so there is no need to remember anything or to hold any token. Each individual can be verified using only distinctive body traits such as features of the face, fingerprints and voice which can be extremely difficult to lose, copy, guess, share or forget. Moreover, the process involved can be easy to use, fast, reliable, and accurate and thus can provide an excellent degree of security compared to other methods. Therefore, biometrics offers a promising approach to the problem of user authentication, and is considered to be more secure, safe and convenient in the field of identity verification.

### **1.1.3 Characteristic Needed by Biometrics Systems**

A good biometric characteristic should satisfy the following ideal criteria, as identified by Jain et al. [1]:

1. Universal: everyone should possess the trait.
2. Unique: no two people should have the same trait.
3. Permanent: the trait should not be changeable and should not change over time or in different environmental conditions.
4. Collectable: the trait should be measurable and obtainable.
5. Acceptable: the trait should be acceptable by people and not annoy them.
6. Circumvention: the trait should be resistant to circumvention.

However, in practical no single biometric will meet all of these requirements [1]. Therefore, a selection of biometrics should be chosen for any possible application based on indicators such as sensor and device availability, computational time required and reliability, cost, sensor size and power consumption. Thereby, practical biometrics systems must have the ability to:

1. Achieve acceptably correct recognition accuracy at a reasonable speed.
2. Be acceptable to the public.
3. Not harm people's health.
4. Be robust against different fraudulent methods.

#### **1.1.4 Types of Biometrics Systems**

Any biometrics devices consist of a reader or sensor which represents the interface between the user and the authentication system. Its function is to scan the biometric trait of the individual, such as fingerprint, iris, facial feature or other characteristic. Then software processes and analyses the scanned trait to extract feature sets that are useful in distinguishing amongst different people, and the actual comparisons are performed. The final part of a biometric system is a computer that runs the software and stores the biometrics database. A typical biometric system operates in three modes:

1. Enrolment mode: captures and stores the biometric information for a new individual.
2. Verification mode: performs a one-to-one comparison and checks if a person is who they claim.
3. Identification mode: performs one-to-many comparisons and verifies identity information in the database and who the person is.

As illustrated in Figure 1.2, the first step is the enrolment of a new person in the system. During this stage, the biometric characteristic from the individual is first captured by the biometric sensor in order to generate a digital presentation of the trait. The sensor has to acquire all of the important information, which is usually in the form of an image, but this may change according to the application and the desired traits. Then, all of the required pre-processing is performed on the raw data, such as enhancement, normalization, segmentation and noise removal. After that, the useful and important features are extracted. This step is critical for successful recognition, since the correct features need to be extracted and selected from the biometric information. Therefore, it is the focus of this thesis. The final step of the enrolment modules is to create a template, which is conducted storing the selected feature vectors as a template in the central dataset of the biometric system or on a smart card issued to the individual, or both of these depending on the application for example ID cards which are given to international travellers by government department.

In the authentication mode of identifications a user provides the biometric trait only to the system, so that the input is just a biometric record. The biometric system recognizes an individual by searching the entire of database templates database for a match with the highest similarity. So, one-to-many comparison is performed against a biometric database in order to determine the identity of an unknown user. The system success in establishing the identity of the user if the comparison of the biometric sample with a template in the database can be completed within a previously set threshold, for example, in scanning a crowd with a camera and using face recognition technology to find a wanted man.

In verification mode, the input is a claimed identity and a biometric record. The system executes a one-to-one comparison of the captured biometric trait with a specific template stored in a central biometric database for that identity, in order to verify where or not the user is the person they claim to be. By providing an individual's ID and biometrics data to the system at the same time, the user activates the system to perform the pre-processing and the feature extracting steps in the same way as in the enrolment stage. After that the system compares the feature data extracted from the biometric trait which has been entered against the stored template of the corresponding ID number, smart card, username or PIN to indicate which template must be used for comparison. Finally, the decision module makes the decision concerning identity. For example, this process is used at the Venerable Bede Church of England Secondary School, Sunderland, UK, where a student enters their card into the iris biometric system at the restaurant and looks into a camera which acquires an image of the student's eye in order to recognise people, then processing them quickly and let them pay for school meals [4]. Similarly, the FacePass system from Viisage is used in point-of-sale verification applications such as ATMs to authenticate that the holder of an ID card or token is the authorized person. The system performs a real-time comparison between a user's face and a stored reference image to confirm the identity of that person, therefore obviating the need for PINs.

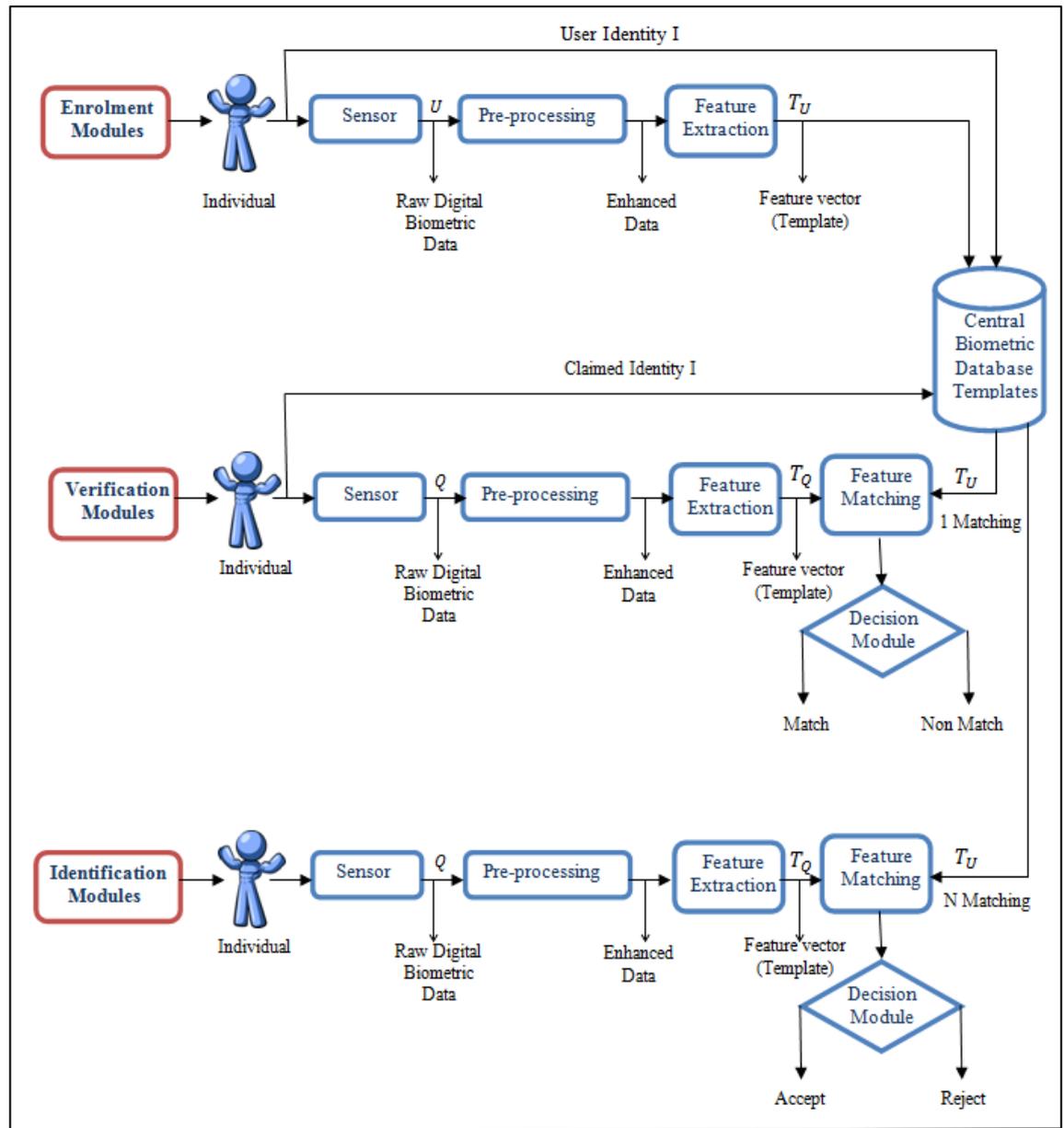


Figure 1.2: Enrolment, Verification, Identification and Matching Modules of a General Biometrics System. Here,  $U$  represent the biometric trait obtained during enrolment,  $Q$  is the query biometric trait obtained during the recognition stages, and  $T_U$  and  $T_Q$  are the biometric template and the query feature sets respectively.

Actually, physiological or behavioural information may change, for example in terms of its angle or shape, when it is read, and so matching scores could change accordingly against the templates in the central biometric database. Therefore, the clearest difference between the traditional authentication system based on an ID card, password or PIN, and the biometrics system is that the latter cannot achieve a 100% match or a non-match response, whilst other existing classical systems can do so according to the exact numbers or letters entered. Accordingly, a person may be wrongly accepted or a valid person may be rejected. The measures developed to evaluate the probabilities of these two cases are called the false accept rate (FAR) and false reject rate (FRR). The FAR represents the probability that the system wrongly matches the input trait to a non-exactly matching template stored in the central biometric database, whereas the FRR represents the probability that the system fails to detect a correct match between the input data and a matching template in the biometric database. There is a trade-off between the errors from both FAR and FRR errors that is typically determined by setting a suitable critical matching value according to the application's purposes. For example, in the case of a very important security system, the possibility that an unauthorized user is accepted should be near zero which is achieved by setting critical matching of a high enough value. But now, even an authorized person may fail to obtain a correct match. On the other hand, if the police are looking for a criminal using face recognition data obtained at the crime scene, it will be important to search for all possible faces. For such a purpose, it is more useful to set critical matching at low enough value to find all possible matches in the face database. In other words, a system with high critical matching value is a high security system but at the same time it is less convenient. Whereas, a system with low critical matching is a less secure system but it is more convenient.

## **1.2 Biometric Facial Recognition**

The face represents the part of body with most obvious and important role in human communication and identification, and so it seems natural to use it for biometric purposes. In fact, facial biometrics is one of the most rapidly growing areas of biometrics, where the image of all face in a photograph or video chip is converted into features that describe the face's characteristics. Compared with other biometric traits, face biometrics systems have the highest levels of collectability and acceptability. They use photographs which are easily acquired. Also, people are willing to accept the use of

images of their face image for identifications in their daily life [3], such as in national identification cards and passports, as shown in Figure 1.3. In addition, facial recognition has a unique property compared to all other biometrics characteristics in that the person can be identified from a distance. It does not require people to place their eye in front of a scanner in a specific position, or their hand on a reader, which processes may not be acceptable in some cultures and sometimes being a source of illness transfer. Face recognition systems instead unobtrusively take pictures of people's faces as they enter a defined area. Furthermore, in most cases people do not feel themselves to be "under surveillance" or that their privacy has been invaded because they are completely unaware of the process. This is also, it is a non-intrusive process and can be completed without any delay to the individual's progress. Therefore, facial biometric recognition is extremely important in surveillance, and has been afforded a higher level of attention by authorities (such as in the U.K., U.S.A., Germany, and France) since the 11<sup>th</sup> of September 2001 because no other biometrics system is so successful in recognizing and determining identity from a distance. One of the most important features of face recognition is that it is very difficult to fool, so an individual cannot easily conceal their identity by using glasses, beards, or even makeup. For example, in 2001 the Tampa Bay Police identified one of the spectators who attended the Super Bowl game against a database of known felons using facial recognition technology [5].



Figure 1.3: Identification (ID) card produced by the Immigration Office in the United Kingdom

However, facial biometrics technology has encountered some challenges which need to be overcome, such as the fact that images of people can be secretly captured with high quality cameras that are hidden in the environment, which leads to privacy concerns. So, perceptions of face recognition have been negatively influenced by the

fear of being monitored by cameras [3]. Furthermore, correct levels of recognition accuracy are not as high as those with other biometrics systems such as iris and fingerprint recognition because data in the latter are usually taken from an exact position in front of the sensor and not from a distance as in facial images. In addition, inaccuracies may occur due to facial wear, and differences in external features such as light conditions and people's bodily movements.

Despite this, advances in technologies such as high quality digital cameras, mobile devices and the internet are increasing the popularity of facial recognition and its performance is improving. These days, facial recognition systems have many applications, such as in biometric authentication, human-computer interaction, commercial security, law enforcement, civil and forensic applications that including access control, image and video surveillance, immigration, national and international ID systems, medical diagnosis and in the entertainment industry [3, 6].

### **1.2.1 Approaches to Face Recognition**

Automated face recognition requires a variety of approaches from different research area to be used, including computer vision, pattern recognition, image processing and machine learning. Most current face recognition approaches use one of two methods of algorithms representation, local feature-based [7-12], and appearance or (holistic)-based [13-16], and their inputs are mostly gray-level images. Feature-based methods identify local face features such as the mouth, nose, eyes and skin irregularities in an individual's image and a representation is created based on their geometric configuration. Meanwhile, appearance-based algorithms use the complete face area and integrate shape and texture information to generate a representation from 2-D images in order to avoid the complexities inherent in 3-D modelling and shape detection.

Although levels of recognition performance for specific applications depend on the determination and extraction of local facial features, the extraction techniques used are not reliable enough for acceptable accuracy in facial recognition. This is because most local feature-based algorithms pre suppose a number of geometric models, and such geometric properties alone are insufficient for face recognition, because important information available in appearance and facial texture is discarded. For example, most eye location techniques assume a number of geometric models of the eyes, and the techniques easily fail if the eyes are closed or glasses are worn. On the other hand, the

appearance-based approach has some important advantages in facial recognition applications, since the representations used are based on intrinsic physical relationships with natural real faces. Furthermore, this approach is less sensitive to variations in illumination and viewpoint compared to the other approaches [17]. Moreover, evidence has shown that the recognition of facial features is dependent on holistic processes involving the interdependence between feature and shape information [2]. Therefore, this approach is the most successful so far in the face recognition field [2, 17-20], and it tends to be more practical, reliable and easier to implement [19]. As a result, this thesis focuses on appearance-based learning, and uses face images that are treated as holistic facial patterns.

### **1.2.2 Challenges**

Although many studies have been conducted into appearance-based face recognition and significantly progress has been achieved, there are still some key technical challenges related to this research field that need to be solved which included the following:

1. There are an extensive range of variations in the human face due to intra-subject differences in facial patterns caused by changes in facial appearance, imaging angle or pose, viewpoints, facial expression, illumination, ageing or over time, articulation, occlusion, and the presence of glasses and makeup. Sometime these variations can be greater than those caused by differences in identity thus it making feature extraction a difficult task.
2. It is commonly believed that such a large number of variations make the distribution of facial features nonlinear dataspace and complex in any space which is linear with respect to the original. Furthermore, in some biometrics and surveillance applications, the variation in camera pose make distinguishing individual faces in the feature space more complicated than with frontal images of faces, which adds further complications to the problem of robust face recognition. This contradicts the current assumption in pattern recognition approaches of much simpler feature distribution, making human face recognition research even more difficult.
3. Usually, the grey or colour face patterns used for recognition have high dimensionality in typical recognition applications. For example, the size of a

grey facial image ranges from  $32 \times 32$  to  $150 \times 130$  pixels [21, 22]. The size of a colour facial image is slightly larger, ranging from  $32 \times 32 \times 3$  to  $150 \times 180 \times 3$  pixels. But, in practical facial recognition applications, the number of training samples per identity or subject is often much smaller than the number of parameters to be estimated, which causes the problem of small sample sizes. This problem often reduces the performance of the feature extractors and classifiers which are essential in supervised learning approaches.

The above challenges present different face recognition methods, some of which are examined in this thesis with an emphasis on the feature extraction module shown in Figure 1.2.

### **1.3 Motivation**

This section begins with simple explanation of the representation of natural facial images. Then popular appearance-based face recognition algorithms are illustrated, focusing on subspace learning algorithms for feature extraction.

#### **1.3.1 Face Image Representation**

Relatively massive multidimensional images of faces are generated in great numbers by various applications. These natural images are the composite consequence of multiple factors and modes related to scene structure, illumination, imaging and viewpoint, and include different facial expressions, head poses, and lighting conditions. The multidimensional objects are formally termed (tensorial data/objects). The expression (tensor) has two different meanings in the fields of physics and mathematics. In this thesis, tensor refers to the same notation used in multilinear mathematics [23], whereas in physics it refers to the generation of a tensor field [24]. So, a tensor also called the  $N$  –Dimensional data/array can be defined mathematically as the generation of vectors and a matrix. As it is well known, when data consists of one dimension ( $N = 1$ ) it is defined as a vector, and when it consists of two dimensions ( $N = 2$ ) it is defined as a matrix, whereas when it consists of more than two dimensions ( $N > 2$ ) its known as tensor. Tensor elements are described with  $N$  indices that define its order, and each index refer to one of its modes [25].

Naturally different objects have some specific structures which take the form of two-, three- or even higher-order tensors [26]. For instance, a grey level face image is a two dimensional tensor object, while a colour image and also the sequence data from grey images in a video are in the form of third-order tensor objects. Thus, algorithms are required to extract useful information from such large data as these data objects in a high-dimensional space. This is a challenging problem because of the variability and complexity of facial pattern distribution, as mentioned in Section 1.2.2. Facial recognition approaches that directly operate on such high-dimensional space usually suffer from the problem known as curse of dimensionality. The essence of this problem is that highly-dimensional data are difficult to handle directly due to the addition of more features that may increase noise and hence increase the resulting error, so decreasing the accuracy rate, in addition to being computationally very expensive [27]. Actually, facial images are highly constrained to a dataspace of intrinsically low dimension or subspace [28]. Therefore, subspace learning or dimensionality reduction aims to map the high-dimensional data set to a lower dimensional set whilst keeping the most distinctive information and preserving the particular properties such as global and local neighbourhood geometric information [29], this is therefore the most successful approach in appearance-based learning, and involves two kinds of subspace learning algorithms: linear- and multilinear algorithm.

### **1.3.2 Linear subspace learning**

Traditional linear subspace learning methods are based on one-dimensional algorithms. Various methods have been used for facial recognition in the last two decades, such as classical principal component analysis (PCA) [15], linear discriminate analysis (LDA) [14], local preserving projection (LPP) [30, 31], orthogonal neighborhood preserving projection (ONPP) and neighborhood preserving projection (NPP)[32, 33]. Whereas PCA searches for the most efficient directions of representation, LDA seeks the efficient direction of discrimination, and LLP is based upon finding the optimal linear approximations to the eigen functions of the Laplace Beltrami operator on the dataspace [33]. On the other hand, the ONPP and NPP algorithms use data-driven weights by solving a least-squares problem to reflect the intrinsic geometry of the local neighborhoods. The ONPP enforces orthogonal mapping and solves an ordinary eigenvalue problem, whereas the NPP imposes a condition of orthogonality on the projected data such that it requires solving as a generalized eigenvalue problem.

Because all of these linear subspace learning techniques are based on one-dimensional methods, as mentioned above, applying the algorithms to tensorial facial data would require the data to be firstly reshaped into a 1-D vector for analysis. This vectorization or reshaping ignores the underlying structure of the data due to being destroyed the natural structure and internal correlations in the original multilinear data, in addition to losing the most useful representation of tensorial facial data. Furthermore, such reshaping usually requires more computational time for training, which often leads to the problem of dimensionality and the evaluation of the covariance matrix may be affected resulting in a reduction of the recognition rate. On the other hand, multilinear subspace learning methods work on natural tensorial objects, and extract features directly from the tensorial representation. Therefore, it is a promising option for the processing of facial tensor objects.

### **1.3.3 Multilinear subspace learning**

In the last few years, multilinear algebra, as the algebra of higher order tensors has received significant attention as many researchers have begun to represent data in their natural form [22, 26, 34-44]. These methods extract features immediately from the tensorial facial data without needing to reshape the object data into one vector. Therefore this huge of algebra has been applied to the analysis of the multifactor structure of images [45]. This has inspired the recent development of multilinear subspace learning algorithms, which extend useful linear subspace learning methods into their multilinear subspace learning form methods such as multilinear principal component analysis MPCA [46], Discriminant analysis with tensor representation (DATER) [26], and multilinear supervised neighborhood embedding ND-TSNE [44] extract features directly from the tensorial facial data instead of the reshaping the data into one long vector, and it has been proven that useful and more compact features can be obtained in this way.

Even with the good progress made in this field, multilinear subspace learning is still in its infancy. And although, the MPCA, DATER and ND-TSNE algorithms have enhanced the performance accuracy of their linear subspace learning forms, but some drawbacks remain Firstly, the MPCA and DATER both preserve the global structure of data samples but at the cost of losing major types of information relevant to the local geometry of their neighbors. In many classification problems, the local structure is more important than the global Euclidean structure, especially when using nearest neighbor

classifier. Secondly, the categorization of the multilinear algorithms mentioned above to show whether they can operate in supervised or unsupervised modes is shown in Table 1.1. Finally, the ND-TSNE method is sensitive to the width of Gaussian envelope selected, which may increase the error rate.

Table 1.1: Categorization of Different Algorithms

Algorithm	Learning Method
DATER [26]	Supervised <sup>1</sup>
ND-TSNE [44]	Supervised <sup>1</sup>
MPCA [46]	Unsupervised

<sup>1</sup> require additional information about the class labels

## 1.4 Contribution

In light of the preceding discussion, this thesis offers five main research contributions as follows:

1. The study introduces a new approach to dimensionality reduction in order to address the problem of bidirectional two-dimensional human facial recognition. The proposed approach called B2DNPP expands the two-dimensional neighborhood preserving projection (2DNPP) method into its bidirectional two-dimensional form. The B2DNPP obtains only one weight matrix to extract two projection matrices separately and sequentially. The 2DNPP performs the reduction only on one direction of a matrix image, while the proposed B2DNPP works on both the row and column directions. Therefore, it preserves the local structure of the dataspace and generates distinctive feature vectors.
2. A new feature framework is presented based on the newly developed B2DNPP algorithm. This is based on feature sets extracted from the curvelet transform, so it is called the C-B2DNPP. The performance of C-B2DNPP is evaluated using an extreme learning machine (ELM) for single-hidden layer feed-forward neural networks. This classifier randomly selects weights of hidden neurons and determines the output weights using an analytical method. The proposed C-

B2DNPP method achieves decreases in error rate more than 40%, 45%, and 27% respectively with the ORL, AR, and FERET databases compared with 2DNPP approach.

3. A novel biometric face recognition approach is developed based on a multilinear neighborhood preserving projection (MNPP) algorithm, which is derived from the original NPP algorithm in order to naturally extend the standard 1-D NPP to the multilinear case. A full mathematical proof of which is given in chapter 5 in order to enhance face recognition performance. This approach naturally represents multidimensional objects as higher-order tensors, and it maps a high-dimensional tensor to a lower dimensional data space by implementing compression in all directions. Therefore, it performs reduction and preserves the hidden relationships in all directions. Besides this, the MNPP can be implemented in either an unsupervised or supervised manner, which may require prior knowledge related to the number of neighbors. Moreover, the MNPP protects the local manifold structure, which is more important than the global Euclidean structure. This is especially relevant in face recognition and data reduction problems. Furthermore, it does not require any selection of parameters during the building of the neighborhood weighting affinity matrix. As well as this, it preserves the inherent geometry of data samples while eliminating noise and redundant data. Therefore, using the suggested approach, MNPP enhances view-based facial recognition and provides levels of recognition performance better than those of all of the methods mentioned in Section 1.3.3. So, it achieved decrease over 50.8%, 75.6%, and 44.6% in error rate using ORL, AR, and FERET databases respectively, compared with 2DNPP.
4. A number of experiments have been performed on three benchmark human facial databases, the ORL, AR, and FERET, in order to evaluate the performance of the proposed B2DNPE and C-B2DNPP algorithms, and to prove the superiority of the proposed algorithms over the existing state-of-the-art methods such as 2DPCA [47], 2DLDA [48], 2DLPP [49], 2DNPP [50], (2D)2PCA [51], (2D)2FLD [52], B2DLPP [53] and Curvelet+B2DPCA[54].
5. Broad comparisons are made between the MNPP and other algorithms, particularly in comparing to the two-dimensional neighborhood preserving projection (2DNPP [50]) with other techniques such as the MPCA[46], DATER [26], ND-TSNE [44] , 2DPCA [47], 2DLDA [48], 2DLPP [49], PCA[15], LDA

[14], NPP [33], and LPP [30]. Empirical identification and verification studies show the power of the MNPP tensor learning and the iterative feature extraction model for facial recognition applications. So, by applying MNPP to human face recognition, state-of-the-art recognition accuracy is achieved.

## **1.5 Organization of the Thesis**

The rest of this thesis is organized as follows:

Chapter 2 introduces some of the linear subspace learning approaches used for feature extraction, and gives a brief review of the dimensionality reduction problem. Then, some popular one-dimensional subspace learning methods that map input images into high-dimensional subspace are explained. Furthermore, the two-dimensional subspace learning methods that operate directly on 2D-images aiming for better recognition performance are illustrated.

Chapter 3 provides an overview of the fundamental concepts of the multilinear subspace central to the implementation of the new multilinear subspace method in this thesis. It starts by introducing the background of MSL as well as basic multilinear algebra along with the tensor distance measure. Then, the tensor-to-tensor multilinear projection is explained. Next, the problems with multilinear subspace learning are defined, followed by a review of existing supervised and unsupervised MSL algorithms highlighting their limitations. Furthermore, an overview is presented of performance evaluation techniques. Finally, three facial databases commonly used in experimental evaluation are described.

Chapter 4 introduces the B2DNPP and the C-B2DNPP methods. It presents a brief introduction to the concepts of the curvelet transform, in addition to giving an overview of existing face recognition techniques that use different feature extraction methods based on multi-resolution tools. Then, the two-dimensional neighborhood preserving projection (2DNPP) method is illustrated followed by a presentation of the problems with bidirectional two-dimensional approach for feature extraction, before a solution using the proposed (B2DNPP) is derived. Then the architecture and design of the ELM classifier is explained. After that, the methodology of the proposed C-B2DNPP facial recognition framework is illustrated. Finally, an extensive study to evaluate the C-B2DNPP against some other 1-D and 2-D methods is presented and discussed. Three set

of experiments are designed and conducted using the three different databases, ORL, AR, and FERET.

Chapter 5 proposes a novel MNPP for face recognition that operates directly on the tensorial data rather than vectors or matrices. This chapter specifies the main difference between the proposed MNPP method and other existing approaches. In addition, it presents the advantages of the MNPP method. Following the MNPP analysis is formulated and its design problem is discussed, which include the initialization of projection, the iterative algorithm, the termination procedure, convergence and issues of projection order. After that, the associated recognition problems are demonstrated. The chapter details a number of experiments conducted in order to evaluate the performance of the MNPP compared with existing algorithms. These experimental studies are performed using the three facial datasets during the two facial biometric modes of identification and verification. Followed a discussion of the results the major findings of this work are summarized.

Chapter 6 concludes this thesis by summarizing the main points of each chapter, and suggesting some directions for future work.

# Chapter 2:

## Linear Subspace Learning

---

Face recognition technology has been an ongoing area of research area for the last 30 years, with significantly increased research activity since 1990 [15]. Nowadays, it has become a routine part of our daily life. It can be assistance in many different areas, including access control, law enforcement, video surveillance, driving licenses, airport security, passports and Immigration, scene analysis, and homeland security. The linear subspace learning approach (LSL) is a popular method of face recognition. It aims to detect the essential features of high dimensional data using linear projection. This chapter surveys some relevant appearance-based algorithms and focuses on the LSL approach to feature extraction. It starts with a brief review of the dimensionality problem, and then some popular one-dimensional subspace learning methods which map the input images into high-dimensional subspace are explained. Finally, it moves on to the two-dimensional subspace learning methods that operate directly on 2D-images for hopefully better recognition performance are discussed.

### 2.1 Introduction

Unlike artificial image recognition systems, people have an amazing ability to recognize faces despite a variety of expressive facial geometries, viewpoints, head poses and lighting conditions [55, 56]. Humans can recognize faces quite easy even when the matching images are distorted, such as when individuals wear glasses or makeup or are occluded by other objects. However, the human brain also has shortcomings in some respects for example; in its capacity to handle large amounts of information and the ability to perform recognition tasks in a repeated way. Therefore, the observation and interpretation of how people perform facial matching is an essential research topic for AI scientists in order to overcome the weaknesses of the system used by the brain. Also, it is a good starting point for automatic attribute analysis in order to aiming to take advantage of the benefits of present technologies and the increasing proliferation of images. These days we are surrounded by multidimensional images of significant size

generated routinely in various applications. These images represent the interaction of multiple factors related to scene structure, illumination and viewpoint, and so on. Therefore, robust algorithms are required if faces are to be recognized and useful information extracted from this huge amount data and in the light of the significant variations in all these factors. Many processes are involved, such as the recognition, classification, segmentation, enhancement of the image data. However, directly operating on images which are commonly specified in a high-dimensional space is difficult. High-dimensional objects present various mathematical challenges, such as the curse of dimensionality mentioned previously [57], the fact that handling high dimensional data objects is computationally expensive, and many classifiers work poorly in high dimensional spaces if only a few training objects are used [58]. It has also been indicated that the recognition system is liable to the problem of over-fitting if the dimensions of the training objects are similar to the size of the training set, which then results in poor performance. Experiments have shown that, when the number of dimensions increases linearly, the required number of examples for learning increases exponentially [59],[60].

Another problem with high-dimensional datasets is that, in many cases, not all the features are relevant to the primary phenomena of interest. More often, only a subset of features is relevant [28]. This problem can be caused by various factors, such as:

1. A lot of dimensions will be correlated with others, leading to redundancy.
2. A lot of dimensions will have variations smaller than the noise, thus it will be irrelevant.

It is important in many facial recognition approaches to remove the redundant and irrelevant dimensions, thus producing a more economical representation of the data [61]. Therefore, the dimensions of the original data should be reduced prior to any modelling process [30]. In addition, the need for reduction in dimensions occurs when the dataset related to high-dimensional representations. The key point here is that, even though facial images can be regarded as objects in a high-dimensional space, they often lie on a manifold of much lower dimensionality embedded in the high-dimensional image space. This leads to the consideration of methods of dimensionality reduction that allow representing the data in a lower-dimensional space. In general, dimensionality reduction approaches aim to remove redundant and irrelevant data in order to avoid data

over-fitting and address to the curse of dimensionality and reduce computational costs. As a result, recognition performance is improve, and the quality of data used in applications such as pattern recognition is enhanced [62].

Dimensionality reduction is a research field at the intersection of different disciplines, including those concerning artificial intelligence, visualization, image processing, pattern recognition, machine learning, data mining, text mining, statistical learning and databases. Each of these disciplines has its own way of looking at the problems involved. For example, in text mining, the problem of dimensionality reduction is defined as selecting a small subset of original words or terms, rather than features. Meanwhile in pattern recognition the problem is defined as extracting and selecting a small subset of features that keep most of the important information from the original dataset but reduce the number of random variables under consideration.

Dimension reduction approaches can be defined in terms of pattern recognition as the process of reducing the number of random features under consideration, so that  $\mathbf{R}^n \rightarrow \mathbf{R}^d$  where  $(d < n)$ [63]. This process can be classified in different ways as follows:

1. Local and global, depending on the feature preserved:  
Global dimensionality reduction approaches are based on preserving the global structure of data samples, while local methods preserve the local structure shared by data samples.
2. Supervised and unsupervised, depending on classification:  
Unsupervised reduction methods do not use any class information, whereas the supervised methods can take advantage of any class information available in the data, and may sometimes need further information regarding classes.
3. Linear and multilinear, depending on the data space used:  
Linear methods need to convert the image into one high-dimensional vector, while multilinear algorithms operate directly on tensorial representations.
4. Linear and nonlinear, depending on the mapping transformation function used:  
The original sample space is transformed by applying either a linear or nonlinear transformation function.
5. Feature extraction and feature selection, depending on processing strategy:

In feature extraction, new features are extracted by applying a mapping function to the linear or multilinear dimensional data space. However in feature selection a subset of the original features are found and selected. This changes the size of the original data space into a smaller space of fewer dimensions. Some data analysis methods such as classification, regression, recognition and image retrieval, can be conducted in the reduced space more accurately than in the original space.

The following section illustrates the basic concepts and key techniques of the two major categories of dimensionality reduction: feature extraction and selection, respectively. Then, some popular dimensionality reduction methods are discussed in the sections subsequent.

### **2.1.1 Feature Extraction**

When the data samples are too large to be processed using an algorithm, they need to be transformed into a typical set of features, which is essential in exploratory data analysis. The procedure of creating new features from existing ones is called image features extraction [64]. This transforms the image data from the data space  $\mathbf{X} \in \mathbf{R}^n$  into a new feature space  $\mathbf{Y} \in \mathbf{R}^n$  which has the same size ( $n$ ) as the original data space.

The purpose of feature extraction is to map data onto a new feature space, which reveals or enhances the class structure of the data for improved visualization. Therefore, it is a special process by which key features of the data are enhanced and extracted.

This problem can be stated mathematically as follows: given the  $n$ -dimensional random vector variable  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^T, \mathbf{X} \in \mathbf{R}^n$ , find an  $n$ -dimensional feature vector  $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n)^T, \mathbf{Y} \in \mathbf{R}^n$  which describes the original dataset accurately depending on some appropriate criteria.

### **2.1.2 Feature Selection Process**

Reducing the dimensionality of the feature extraction space is an important step in pattern recognition tasks. The process of using a small subset of the features of an image by considering the intrinsic characteristics of each object feature, such as the variance, correlation, angles, distances, or clustering), is called image feature selection.

Here a subset  $\mathbf{Y} \in \mathbf{R}^d$  is chosen of all the features  $\mathbf{Y} \in \mathbf{R}^n$ . Feature selection enhances the performance of learning models, by removing the most irrelevant and redundant features from the data [63]. In addition it improves the understanding of the extracted features by identifying important features and determining how they are related to each other.

In mathematical terms, the feature selection problem can be formulated as follows: given the  $n$ -dimensional feature vector  $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n)^T, \mathbf{Y} \in \mathbf{R}^n$  find a smaller dimension  $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_d)^T, \mathbf{Y} \in \mathbf{R}^d$  where  $(d < n)$ . This captures the distinctive information in the original data space according to some criterion. The variable  $d$  has different names depending on the applications using it. For example, it is commonly known as a feature or attribute in the pattern recognition and machine learning fields, while the term variable is used in statistics.

In general, the feature extraction process recombines the exciting original information and converts it into new features, using a linear or multilinear mapping transformation function. So it transforms any original  $n$ -dimensional object vector into a  $n$ -dimensional feature vector:

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^T \xrightarrow{\text{Feature extraction}} \mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n)^T$$

On the other hand, feature selection is a process that converts the  $n$ -dimensional feature vector to a lower  $d$ -dimensional features vector. Therefore, it ideally preserves or even enhances the distinctive information while simultaneously reducing the dimensionality of the feature vector:

$$\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n)^T \xrightarrow{\text{Feature selection}} \mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_d)^T, d < n$$

In either case, the aim of dimensionality reduction is to find a low-dimensional representation of the data which still describes the data with sufficient accuracy [65].

## 2.2 Linear Subspace Learning Approach

For reasons of computational and conceptual simplicity, a representation which is a linear transformation of the original data is often preferred. Therefore, dimensionality

reduction algorithms using linear subspace learning (LSL) methods have attracted considerable interest in recent years [66]. They have been very popular in determining the intrinsic dimensionality of a data manifold, in addition to extracting its principal directions. As a commonly used feature extraction and dimensionality reduction technique, it also provides a solid foundation for a variety of nonlinear approaches. This is evidenced by various approaches published in the past few decades. This technique has been successfully used in various computer vision, information retrieval and pattern recognition applications, such as appearance-based face recognition.

In the well-known linear subspace learning (LSL) approach, each image in the data space acts as a linear combination of its original data. In the classical LSL methods used for feature extraction, the facial image is represented as a vector in high-dimensional space. However, any face image is essentially a second- or third- order tensor, and so it has to be transformed into a vector in vector representation. A typical technique used to conduct this process is so-called vectorization or matrix-to-vector alignment which concatenates all the rows or columns in the matrix together to give a long vector.

### **2.2.1 LSL problem definition**

Mathematically, a classical linear transformation problem can be stated as follows. Let  $\{\mathbf{X}\}$  be a facial dataset of  $M$  training objects ( $\mathbf{X} \in \mathbf{R}^{\mathbf{I} \times \mathbf{M}}, m = 1, 2, \dots, M$ ). Each object  $\mathbf{x}_m$  is a single vector of size  $(\mathbf{I} \times 1)$  in a vector space  $\mathbf{R}^{\mathbf{I} \times 1}$ . The objective of LSL is to define the linear transformation projection matrix  $\mathbf{U} \in \mathbf{R}^{\mathbf{I} \times \mathbf{P}}$ , which maps the original dataset into a smaller set  $\mathbf{Y} \in \mathbf{R}^{\mathbf{P} \times \mathbf{M}}$  with  $(\mathbf{P} < \mathbf{I})$ , such that each object  $\mathbf{x}_m$  can be represented directly by  $\mathbf{y}_m$  using the transformation function  $\mathbf{y}_m = \mathbf{U}^T \mathbf{x}_m$ . So, the extracted features or projected objects  $\{\mathbf{y}_m\}$  satisfy an optimality criterion [67]. Then, in the classification step, these features  $\{\mathbf{y}_m\}$  are fed into a classifier such that the nearest neighbor classifier and the similarity decision is usually considered based on some distance measure.

### **2.2.2 Motivation for the use linear-based algorithms**

In the last two decades, linear subspace learning approaches such as PCA [15] and LDA [14, 68] have been widely used for feature extraction or selection, which includes

dimensionality reduction, and they have been proven to be efficient for modeling or classification [57]. These two methods are the most widely used subspace learning techniques for face recognition. The training object faces are projected into a low dimensional representation space where recognition is performed. The main idea behind this methodology is that face space given by the feature vectors has lower dimensions than the image space that given by the number of pixels in the original raw image, so that the subsequent recognition of faces can be performed in this reduced space. PCA has the benefit of capturing holistic features, but at the same time ignore localized features. Whereas fisher faces from LDA technique extracts features which can be used to discriminate between classes and is found to perform better for large datasets. Its main shortcoming, however, is that of small sample space (SSS). While PCA searches for efficient directions of representation, LDA searches for the projection axes on which the data points of different classes are far from each other while requiring the data points of the same class to be close to each other. Unlike PCA, which determines information in an orthogonal linear space, LDA encodes discriminating information in a linear separable space using bases that are not necessarily orthogonal. It is generally believed that algorithms based on LDA are superior to those based on PCA in the context of pattern classification, but the former is more susceptible to the problem of small sample size. However, a number of recent studies show that when the number of training samples per class in the facial dataset is much smaller than the dimensionality of the input sample space, or the if training samples are not representative of those in the test, PCA can outperform LDA.

Projection in both PCA and LDA is based on the global structure of the data samples. This may lead to the loss of important information about the local geometry within each neighborhood. But in many classification problems the local manifold structure is more important than the global Euclidean structure, especially when nearest neighbor classifiers are used. Both PCA and LDA effectively observe only the Euclidean structure. Therefore, they fail to discover the underlying structure if the face images lie on a nonlinear submanifold hidden in the image space. Recently, therefore, significant attention has been devoted to geometrically motivated approaches to visual analysis. Some nonlinear techniques have been proposed in order to discover the nonlinear structure of the manifold such as isometric feature mapping (Isomap) [69], the Laplacian Eigenmap (LE) [70], and locally linear embedding (LLE) [71]. These

nonlinear approaches all use local neighborhood relationships to learn the global structure of nonlinear manifolds, but they have quite different motivations and derivations. However, although they do yield impressive results when used on certain benchmark artificial data sets, they have limitations such as an inability to map a new testing image directly. Instead this is defined only for the training data points, so how the result maps to new test points remains unclear. Furthermore, their expensive computation costs are significantly higher than most linear dimension reduction methods. Therefore, these methods may result in an Out of Sample problem [15]. To overcome this problem, He et al. [30, 31] designed a local preserving projection (LPP) algorithm which derives definitions comprehensively training and test data points. This algorithm was proposed based on the Laplacian Eigenmap (LE) and its output are called Laplacian faces [31]. LLP is a method for linear dimension reduction which is based upon finding the optimal linear approximations to the eigen functions of the Laplacec Beltrami operator on the manifold [30]. Recently, Yanwei et al. [33] and Kokopoulou and Saad [72] have implemented two algorithms, called neighborhood preserving projection (NPP) and orthogonal neighborhood preserving projection (ONPP) respectively, which operate in a way similar to the LLE method. These approaches preserve the local structure between samples and also result in acceptable rates of facial recognition. These algorithms use data-driven weights by solving a least-squares problem to reflect the intrinsic geometry of the local neighborhoods. ONPP enforces mapping as orthogonal and solves an ordinary eigenvalue problem, whereas the NPP imposes a condition of orthogonality on the projected data, and thus requires the solution of a generalized eigenvalue problem.

### **2.3 Relevant Previous Work on LSL**

Among all the different ways of categorizing dimensionality reduction methods mentioned above, this section briefly describes four popular forms of classical feature extraction using the LSL techniques: PCA, LDA, LPP and NPP. These are discussed in terms of feature extraction or selection, local or global mapping, and supervised or unsupervised learning.

### 2.3.1 Principle Component Analysis (PCA)

The well-known eigenface method for face recognition proposed by Turk and Pentland [15] is also called the Karhunen-Loeve transform in the signal processing literature, and is one of the most influential linear subspace learning methods. Motivated by the earlier work of Sirowich and Kirby [73, 74], eigenface recognition approach derives its name from the German prefix *eigen*, which means individual or own. This method of facial recognition [15] is considered to be the first working in the field of facial recognition technology [75]. Built on PCA, it initialled the era of appearance-based approaches to human face recognition, and more generally of visual object recognition. The idea behind PCA is to reduce the dimensionality of a dataset consisting of a large number of correlated variables, while retaining as much as possible the variation present in the original dataset. This is accomplished through the transformation of the interrelated variables of the dataset to a new set of uncorrelated variables called principal components (PCs). The PCs are non-zero orthogonal vectors that maximally capture the relationship between the original dimensions, so that the first few PCs retain most of the original variation in the data [66]. Therefore, PCA aims to preserve the most distinctive features by finding a set of mutual orthogonal functions that capture the directions of maximum variance in the dataset.

Turk and Pentland [15] treated the face images as a 1-D vector and worked on the image as a whole. They used a nearest mean classifier to classify different face images. They obtained an approach for face recognition which depends on the monitoring that the projections of a face image and a non-face image are completely different. They applied the method with a database consisting of 2500 face images from 16 subjects, in combinations of 3 head sizes, 3 head orientations and 3 lighting conditions. They performed experiments in order to test the robustness of their method against variations in size, head orientation, illumination, and the differences between training and test conditions. Although the system was quite robust to changes in illumination, performance declined rapidly as the scale changed [15]. This can be explained in terms of the correlation between facial images obtained under different illumination conditions, where the correlation between face images at different scales is actually quite low. In addition, Turk and Pentland [15] reported that the eigenface recognition approach performed well as long as the test image is similar to the training images used for obtaining the eigenfaces.

In the eigenface recognition technique as shown in Figure 2.1, variation is calculated by the total degree of scatter of all feature vectors related to the mean vector through the total covariance matrix  $\mathbf{C}$  which is defined as follows:

$$\mathbf{C}_{COV} = \frac{1}{M} \sum_{m=1}^M (\mathbf{x}_m - \bar{\mathbf{x}})(\mathbf{x}_m - \bar{\mathbf{x}})^T, \quad (2.1)$$

$$\bar{\mathbf{x}} = \frac{1}{M} \sum_{m=1}^M \mathbf{x}_m, \quad (2.2)$$

where  $\bar{\mathbf{x}}$  is the mean face vector of all training facial images in the dataset. Then, compose the PCA projection matrix  $\mathbf{U}_{PCA}$  is constructed to consist of the  $P$  eigenvectors as columns corresponding to the largest  $P$  eigenvalues of the scatter matrix  $\mathbf{C}$ , where ( $P < I$ ). The objective function of PCA is as follows:

$$\mathbf{U}_{PCA} = \arg \max_{\mathbf{U}} \left| \mathbf{U}^T \mathbf{C}_{cov} \mathbf{U} \right| \quad (2.3)$$

Hence, PCA solves the eigenproblem as:

$$\mathbf{C}_{cov} \mathbf{U} = \lambda \mathbf{U}. \quad (2.4)$$

The original feature vectors can be reconstructed and transformed into the PCA feature space of extracted samples using the transformation function  $y$  as follows:

$$y = \mathbf{U}_{PCA}^T \mathbf{x}_c \quad (2.5)$$

where  $\mathbf{x}_c = (\mathbf{x} - \bar{\mathbf{x}})$  the mean centred image of  $\mathbf{x}$ . The new feature vectors are orthogonal to each other and there is no covariance between any two vectors. The transformation function  $y$  is also used to find the projection of a new test object  $x$  in the PCA feature space. Therefore, PCA can be used as feature extraction method to conserve information, eliminate correlation and reduce dimensionality.

The PCA is an unsupervised learning method, so it does not take any information about class structure into account, even when such information is available. However,

large intra-class variations have a negative impact on classification performance, but both intra-class and inter-class variations are maximized in the PCA feature space. Therefore, for facial classification purposes, it is generally believed that the PCA cannot perform as well as supervised learning techniques such as the LDA [76].

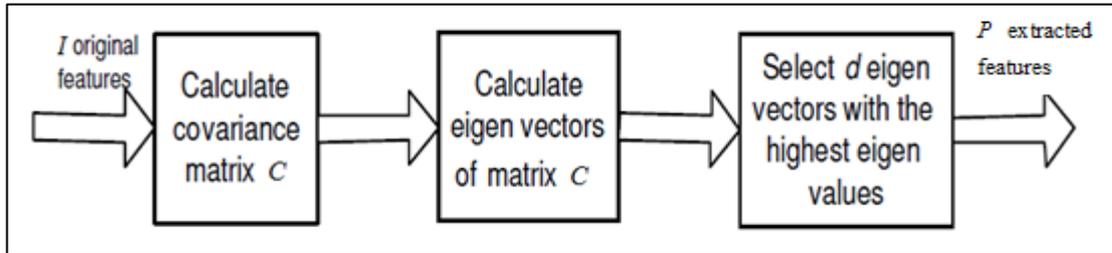


Figure 2.1: The Block diagram of Eigenface Recognition Technique

### 2.3.2 Linear Discriminant Analysis (LDA)

Etemad and Chellappa [68] presented a method which applies on Linear/Fisher Discriminant Analysis (LDA) to the face recognition process. This is a classical supervised linear subspace learning method which has been proven to be an effective approach in developing practical face recognition systems. Unlike PCA, LDA considers class information, and search for projection axes along which the data samples of different classes are further apart from each other, whereas the data samples of the same class are closer to each other. The aims is to derive the most discriminative features by finding the optimal projection that maximizes the between-class (inter-class) variance of the face data but minimizes within-class (intra-class) variance as illustrated in Figure 2.2 . The major disadvantage of LDA is that it is unable to make use of unlabeled points. Moreover, LDA may suffer from the singularity problem when there are insufficient training examples [77].

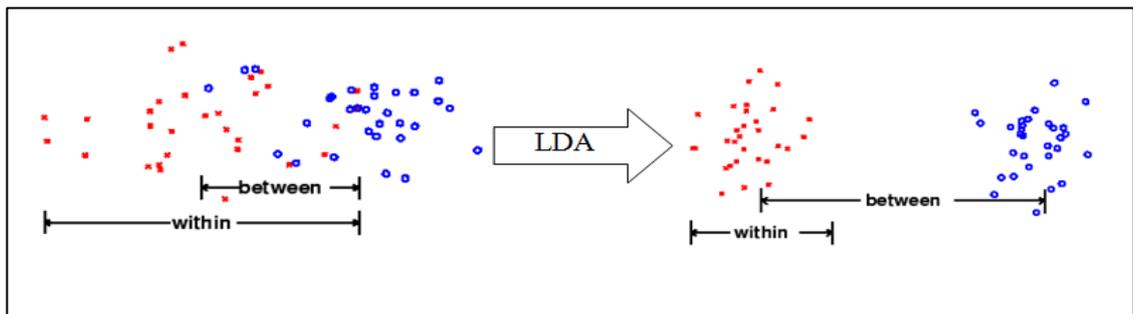


Figure 2.2: Between-class variance and within-class variance before and after using LDA

Figure 2.3 depicts the principal procedure of LDA approach. LDA produces a class specific feature space based on the maximization of a popular separability criterion known as Fisher’s discriminant criterion, which is defined as the ratio of between-class scatter  $\mathbf{S}_B$  to within-class scatter  $\mathbf{S}_W$  achieved by maximizing the following objective function:

$$\mathbf{U}_{LDA} = \arg \max_{\mathbf{U}} \frac{|\mathbf{U}^T \mathbf{S}_B \mathbf{U}|}{|\mathbf{U}^T \mathbf{S}_W \mathbf{U}|} \quad (2.6)$$

where:

$$\mathbf{S}_B = \sum_{c=1}^C M_c (\bar{\mathbf{x}}_c - \bar{\mathbf{x}})(\bar{\mathbf{x}}_c - \bar{\mathbf{x}})^T, \quad (2.7)$$

$$\mathbf{S}_W = \sum_{m=1}^M (\mathbf{x}_m - \bar{\mathbf{x}}_{c_m})(\mathbf{x}_m - \bar{\mathbf{x}}_{c_m})^T. \quad (2.8)$$

In the above equations:

$\mathbf{U}$  is the linear projection matrix

$C$  is the number of classes in the facial dataset,

$c$  is the index of the class,

$c_m$  is the class label for the  $m$ th training image sample,

$M_c$  is the number of training images in class  $c$  and

$\bar{\mathbf{x}}_{c_m}$  is the mean of class  $c$  which is defined as:

$$\bar{\mathbf{x}}_c = \frac{1}{M} \sum_{m, c_m=c} \mathbf{x}_m \quad (2.9)$$

As in the case of PCA, where the eigenfaces are calculated by the eigenvalue analysis, the projections of LDA are calculated by the generalized eigenvalue equation. Therefore, the maximization of Equation 2.6 leads to the generalized eigenvalue problem as follows:

$$\mathbf{S}_B \mathbf{u}_p = \lambda \mathbf{S}_W \mathbf{u}_p, \tag{2.10}$$

where the value of  $\lambda$  is the eigenvalue solution to the minimization problem. Corresponding to the  $P$  largest eigenvalues,  $\mathbf{U}_{LDA}$  is the optimal transformation matrix which consists of the  $P_n$  generalized eigenvectors, which can be used to project each testing sample object into LDA form using the following transformation function:

$$\mathbf{Y} = \mathbf{U}_{LDA}^T \mathbf{X} \tag{2.11}$$

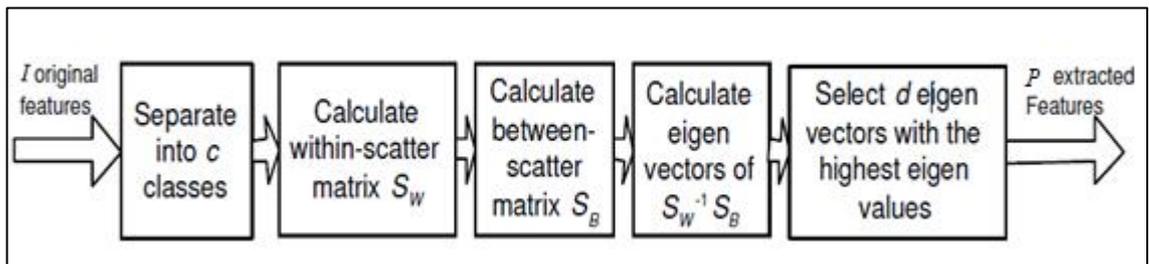


Figure 2.3: The Block Diagram of the LDA Linear Subspace Learning Approach.

Linear discriminant analysis (LDA) and principal component analysis (PCA) are extraction methods based on appearance using the features of the global structure in the dataset space. Using only the global structural features has the disadvantage that the local structural features can not be characterized. However, the methods of locality preserving projection (LPP) and neighborhood preserving projection (NPP) are appearance extraction techniques which preserve local structure features, which is more important than the global Euclidean structure in many classification problems especially in face recognition.

### 2.3.3 Locality Preserving Projection (LPP)

Although, the nonlinear subspace learning techniques for dimensionality reduction such as Isomap, LE and LLE successfully identify complex data manifolds such as the Swiss roll, because cost functions are minimized by local nonlinear dimensionality reduction

methods which aim to preserve local properties of the data. However, in many subspace learning settings, the use of a linear methods for dimensionality reduction is more desirable; for example, when data has to be transformed back or reconstructed into its original space or when an accurate and rapid out-of-sample extension is necessary. However, the classical linear approaches such as PCA effectively preserve the global structure of the facial image space, and LDA preserves the discriminating information. Unlike these approaches and distinct from nonlinear dimensionality reduction methods, He & Niyogi [78] proposed an unsupervised linear method that so-called locality preserving projection (LPP). This combines the benefits of linear and nonlinear approaches to dimensionality reduction by finding a type of linear mapping that minimizes the cost function of the Laplacian eigenmaps (LE). This mapping is defined everywhere in ambient space rather than only on the training data points. The LPP seeks to preserve the local neighborhood structure and intrinsic geometry of the image space by preserving locality. In many real-world classification purposes, the local manifold structure is more important than the global Euclidean structure.

Similar to Laplacian eigenmaps, LPP starts with the construction of a graph incorporating the neighborhood information of the data set, which can be defined in two different ways as follows:

$$1. \text{ Heat kernel: } W_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{t}}, & \text{if } \|x_i - x_j\|^2 < \varepsilon \text{ (if } x_i \text{ and } x_j \text{ are "close") } \\ 0 & \text{otherwise} \end{cases} \quad (2.12)$$

$$2. \text{ Simple-minded : } W_{ij} = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ are "close",} \\ 0 & \text{otherwise} \end{cases} \quad (2.13)$$

Where  $t$  is the width of Gaussian envelop and  $\varepsilon$  is a small constant number. If  $\|x_i - x_j\|^2 < \varepsilon$ , or  $x_i$  is one of the  $k$  nearest neighbors of  $x_j$ , then  $x_i$  and  $x_j$  are close to each other. LPP places an edge between nodes  $i$  and  $j$  if  $x_i$  and  $x_j$  are "close". This method employs the same objective function as in Laplacian eigenmaps:

$$\min \sum_{ij} (y_i - y_j)^2 w_{ij} . \quad (2.14)$$

Subsequently, the minimization in Equation 2.14 leads to the derivation of the transformation vector, which can be accomplished by solving a generalized eigenvalue problem that is given by:

$$\mathbf{XLX}^T \mathbf{U} = \lambda \mathbf{XDX}^T \mathbf{U}. \tag{2.15}$$

Here  $D$  is a diagonal matrix whose entries are columns (or row a  $W$  is symmetric) sum of  $W$ ,  $D_{ii} = \sum_j w_{ij}$ , while  $L$  is the graph Laplacian,  $L = D - W$ . It can be shown that the eigenvectors  $u_i$  corresponding to the  $d$  smallest nonzero eigenvalues form the columns of the linear mapping  $\mathbf{U}$  that minimizes the Laplacian eigenmap cost function. The low-dimensional data representation is given by:

$$\mathbf{Y} = \mathbf{U}_{LPP}^T \mathbf{X} \tag{2.16}$$

Gaussian weights are used in LPP in order to amplify the neighborhood structure and preserve it in the reduced space. But this approach is sensitive to parameters due to the determination of weights needs of an suitable value of the width of the Gaussian envelope [72].

**2.3.4 Neighborhood Preserving Projections (NPP)**

Similarly to LPP, neighborhood preserving projection (NPP) [33] aims to preserve the local neighbourhood structure of the data space. It minimizes the cost function of a local nonlinear technique for dimensionality reduction under the constraint that the transformation from the high-dimensional to low-dimensional data representation involves linear mapping. As the LPP is a linear approximation to Laplacian Eigenmap (LE), NPE is a linear approximation to locally linear embedding (LLE) [71]. The NPE defines a neighborhood graph for the facial dataset, and subsequently computes the reconstruction weights. Therefore, a data point  $x_i$  along its  $k$  neighbors is locally linear on the underlying manifold, and each data point  $x_i$  in the dataset can be reconstructed from its  $k$  weighted neighbors [33, 71]. The reconstruction error is minimized by the cost function:

$$\delta(\mathbf{W}) = \sum_i \left\| x_i - \sum_j w_{ij} x_j \right\|_2^2, \quad (2.17)$$

where the weight  $w_{ij}$  denotes the contribution of the  $j$ th data point to the  $i$ th reconstruction. Thus, each data sample  $x_i$  is reconstructed by a linear combination of its  $k$  nearest neighbor points.

The weights  $w_{ij}$  are subject to the following constraints:

1.  $w_{ii} = 0, i = 1, 2, \dots, M$ ,  $M$  is the total number of images in the facial dataset.
2.  $w_{ij} = 0$ , if  $w_{ij}$  is not one of the  $k$  neighbors of  $x_i$ .
3.  $\sum_j w_{ij} = 1, j = 1, 2, \dots, k$  so  $x_i$  is reconstructed by a combination of its neighbors.

Let  $\mathbf{G} \in R^{K \times K}$  denote the local Grammian matrix at  $x_i$ , and define its entries are defined by the inner product of:

$$\mathcal{G}_{uv} = (s_i - s_u)^T (s_i - s_v) \in R^{K \times K}. \quad (2.18)$$

Let  $\mathbf{S}^{(i)}$  be a system of stacked vectors with respect to  $x_i$  and its neighbors. In order to find the optimal weights  $w_{i,:}$ , the least-squares  $(\mathbf{S}^{(i)} - s_i e^T) w_{i,:} = 0$  needs to be solved subject to the constraint  $e^T w_{i,:} = 1$ , where  $e$  is the vector such that  $e = [1, \dots, 1]^T$ . The suitable solution to this constrained least-squares problem using the inverse of  $\mathbf{G}$  is given by the following [33, 71, 79]:

$$w_{i,:} = \frac{\mathbf{G}^{-1} e}{e^T \mathbf{G}^{-1} e}. \quad (2.19)$$

Determining the weights  $w_{ij}$  with for a given data point  $x_i$  is a simple-minded or local calculation, because it only involves  $x_i$  and its  $k$  neighbors [33, 79]. The weights  $w_{ij}$  are invariant to rotation, scaling and translation. As a result, they preserve the intrinsic geometry of the underlying manifold.

The NPP approach employs the following objective function:

$$\varphi(\mathbf{Y}) = \sum_i \left\| \mathbf{y}_i - \sum_j w_{ij} \mathbf{y}_j \right\|_2^2. \quad (2.20)$$

A constraint  $\frac{1}{M} \sum_i \mathbf{y}_i \mathbf{y}_i^T = \mathbf{I}$ , is imposed on the objective function in order to remove an arbitrary scaling factor in projection. The objective function in Equation 2.20 can be rewritten by employing the following algebraic steps:

$$\varphi(\mathbf{Y}) = \sum_i \left\| \mathbf{y}_i - \sum_j w_{ij} \mathbf{y}_j \right\|_2^2 = \left\| \mathbf{Y} (\mathbf{I} - \mathbf{W}^T) \right\|_F^2 = \text{tr} \left[ \mathbf{Y} (\mathbf{I} - \mathbf{W}^T) (\mathbf{I} - \mathbf{W}^T) \mathbf{Y}^T \right]. \quad (2.21)$$

Using the linear transformation function  $\mathbf{Y} = \mathbf{U}_{NPP}^T \mathbf{X}$ , then Equation 2.21 will become:

$$\varphi(\mathbf{Y}) = \text{tr} \left[ \mathbf{U}_{NPP}^T \mathbf{X} (\mathbf{I} - \mathbf{W}^T) (\mathbf{I} - \mathbf{W}^T) \mathbf{X}^T \mathbf{U}_{NPP} \right]. \quad (2.22)$$

Hence, the NPP minimization problem becomes:

$$\arg \min_{\mathbf{U}^T \mathbf{X} \mathbf{X}^T \mathbf{U}} \mathbf{U}_{NPP}^T \mathbf{X} (\mathbf{I} - \mathbf{W}^T) (\mathbf{I} - \mathbf{W}^T) \mathbf{X}^T \mathbf{U}_{NPP}, \quad (2.23)$$

where  $\mathbf{I}$  is the identity matrix of order  $k$ , and  $\mathbf{U}$  is the transposition of the matrix. The optimal projection axis  $\mathbf{U}$  is given by the minimum eigenvectors solution to the generalized eigenvalue problem as follows:

$$\mathbf{X} (\mathbf{I} - \mathbf{W}^T) (\mathbf{I} - \mathbf{W}^T) \mathbf{X}^T \mathbf{U}_{NPP} = \lambda \mathbf{X} \mathbf{X}^T \mathbf{U}_{NPP}, \quad (2.24)$$

where  $\lambda$  is the eigenvalue solution to the optimization problem. The translation matrix  $\mathbf{U}$  is the basis of the eigenvectors corresponding to the  $d$  smallest generalized eigenvalues. The minimization problem in Equation 2.23 turns into the eigenvalue

problem under the above constraint. If  $YY^T = I$ , the projection is a neighborhood preserving projection (NPP), or otherwise if  $UU^T = I$ , it is called an orthogonal neighborhood preserving projection (ONPP). The new data sample can be transformed from the data space into a reduced space by the transformation function  $Y = U_{NPP}^T X$ .

Table 2.1 summarized the algorithm of the NPP approach, which is consists of four steps as explained below.

Table 2.1: Algorithm of the NPP Approach

1. Find the  $k$  nearest neighbors of each data point  $x_i$ .
2. Calculate the weights  $w_{ij}$  that best reconstruct each data point  $x_i$  from its  $k$  nearest neighbors, solving the constrained least-squares problem in equation.
3. Calculate the linear transform matrix  $U$  by solving the generalized eigenvalue problem
4. Calculate the vectors  $y_i$  that are reconstructed by the weights  $w_{ij}$ , so that dimension reduction is performed. An outline of the NPP algorithm is illustrated in Figure 2.4.

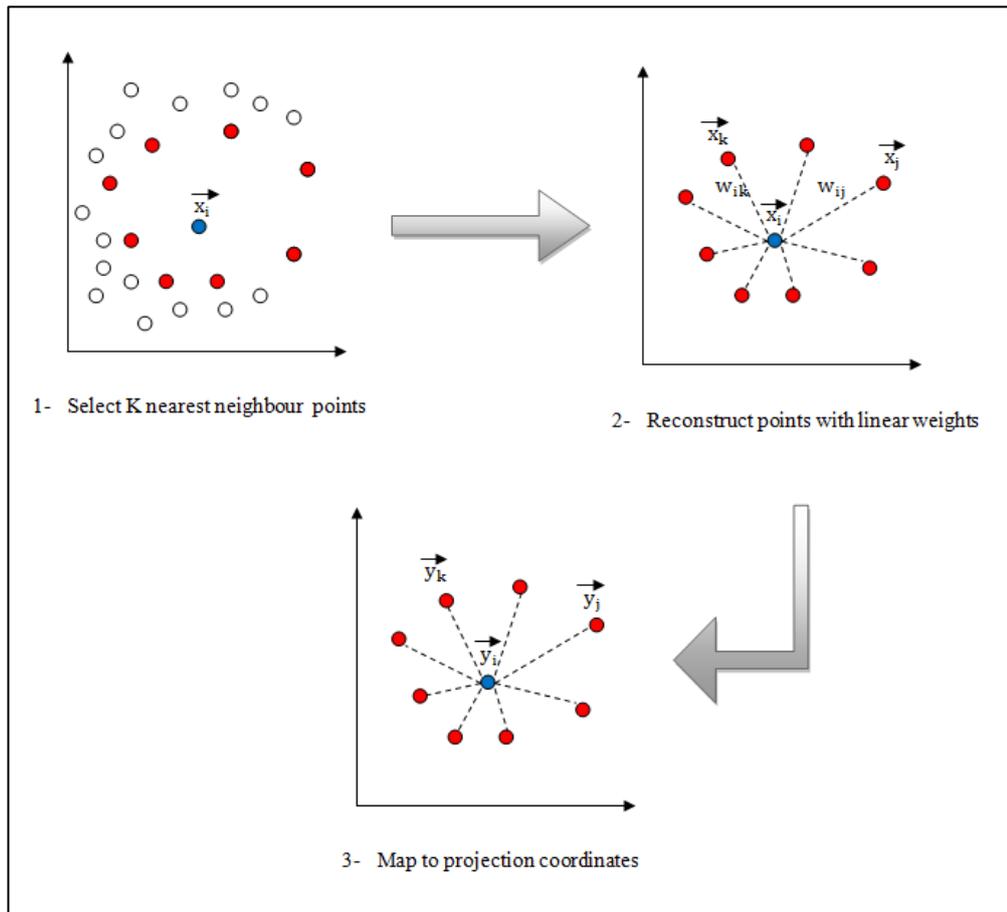


Figure 2.4: Outlines of the NPP approach

Table 2.2 summarizes the approaches to linear feature extraction mentioned above. All of the traditional dimensionality reduction and feature extraction methods, such as PCA, LDA, LPP and NPP were designed to treat sample data objects in the form of one-dimensional vectors. Therefore, tensorial data samples need to be vectorized before these methods can be applied. This vectorization breaks the higher order dependencies present in the natural data structure that can potentially lead to more compact and useful representations [46]. However, potential problems include larger memory requirements, higher computational costs, the limited availability of projection directions, and a loss of information about the underlying spatial structure of the images. In order to overcome these limitations, new methods have been introduced that are directly applicable to two-dimensional data, as explained in the next section.

Table 2.2: Summary of Common LSL Methods for Linear Feature Extraction

Method	Weight matrix	Objective function	Generalized eigenvalue problem
PCA	$C_{cov} = \frac{1}{M} \sum_{m=1}^M (x_m - \bar{x})(x_m - \bar{x})^T$	$\arg \max_U  U^T C_{cov} U $	$C_{cov} U = \lambda U$
LDA	$S_B = \sum_{c=1}^C M_c (\bar{x}_c - \bar{x})(\bar{x}_c - \bar{x})^T,$ $S_W = \sum_{m=1}^M (x_m - \bar{x}_{c_m})(x_m - \bar{x}_{c_m})^T.$	$\arg \max_U \frac{ U^T S_B U }{ U^T S_W U }$	$S_B u_p = \lambda S_W u_p$
LPP	$w_{ij} = \begin{cases} \frac{\ x_i - x_j\ ^2}{t}, & \text{if } \ x_i - x_j\ ^2 < \varepsilon \\ 0 & \text{otherwise} \end{cases}$	$\min \sum_{ij} (y_i - y_j)^2 w_{ij}$	$XLX^T U = \lambda XD X^T U$
NPP	$w_{i,:} = \frac{G^{-1} e}{e^T G^{-1} e}.$ $\mathcal{G}_{uv} = (s_i - s_u)^T (s_i - s_v) \in R^{K \times K}.$	$\min \sum_i \left\  x_i - \sum_j w_{ij} x_j \right\ _2^2$	$X (I - W^T) (I - W^T)^T X^T U_{NPP}$ $= \lambda X X^T U_{NPP}$

## 2.4 Two-Dimensional Linear Subspace Learning Approaches

There are two types of two-dimensional LSL methods. The two-dimensional based algorithms employ compression only in the row direction, and bidirectional two-dimensional compression methods perform compression in both row and column directions.

### 2.4.1 Two-Dimensional LSL Based Methods

In order to overcome the problems of the classical one-dimensional methods, researchers have tried to process the image as a 2D matrix rather than a vector, so that the image matrices do not need to be transformed into vectors. Yang et al. [47] proposed a two-dimensional 2D-PCA algorithm that directly computes the image scatter matrix from the original image matrix representations. Inspired by 2D-PCA, Li and Yuan [48] presented a 2D-LDA which is an extension of traditional LDA using the idea of image matrix representations. Chen et al. [49] and H. Zhang et al. [50] developed 2D-LPP and

2D-NPP algorithms respectively, which directly extract the appropriate features from image matrix representations by preserving the local structure of the samples. It has been reported in [47, 80] that the two-dimensional methods evaluate the covariance image matrix more accurately and compute the corresponding eigenvectors more efficiently than one-dimensional methods. As a result, the recognition accuracy rates with several face databases are significantly improved since the relationships among patterns is retained in the reduced feature space. Also, the extraction of image features is computationally more efficient using two-dimensional methods. Mathematically, a two-dimensional transformation problem can be stated as follows: suppose that  $\{\mathbf{X}\}$  is a facial dataset consisting of  $M$  training 2-D image samples ( $x \in R^{I_1 \times I_2}$ ). The objective of the 2-D LSL approach is to define the linear transformation projection matrix  $\mathbf{U} \in R^{I_2 \times P}$  which maps the original dataset into a smaller set  $Y \in R^{I_1 \times P}$  with ( $P < I_2$ ). Projecting each object  $x_m$  onto  $\mathbf{U}$  yields a feature matrix  $y_m$  of size  $(I_1 \times P)$  using the transformation function  $y_m = x_m \mathbf{U}$  [47]. After that, the extracted features  $\{\mathbf{y}_m\}$  are fed into a classifier in the classification step, and the decisions on similarity are usually made based on some measure distance.

However, the main drawback of 2-D approaches is that they require many more coefficients for image representation than any standard 1-D based approach. This is because the 2-D methods perform compression only in row direction. For example, if the size of 2-D facial images is  $200 \times 200$ , then the number of coefficients according to the 2-D methods is  $200 \times d$ , where  $d \geq 5$ . The corresponding total number of coefficient used in any 1-D based methods is  $d$  only. This problem can be alleviated using the bidirectional two-dimensional approach, which simultaneously considers reduction in both row and column directions.

### 2.4.2 Bidirectional Two-Dimensional Compression Methods

In general, bidirectional two-dimensional methods employ double-sided transformations, performing compression not only in the row direction, but also in the column direction. Recently, researchers have introduced bilateral compression methods such as (2D)2PCA [51], (2D)2FLD [52] and B2DLPP [53], which improve upon previous two-dimensional methods by preserving the internal relationships between image rows and columns. So, 2-D facial matrices can be represented by fewer

coefficients. As a result, bidirectional two-dimensional algorithms have many important advantages over simple two-dimensional methods. Firstly, they show the hidden relations between image row vectors and between image column vectors. Secondly, image feature extraction is simpler and it is easier to evaluate the covariance matrix accurately, so that less computational time is required. Furthermore, the performance of bidirectional two-dimensional methods is usually better than that of 2-D based methods.

In mathematical terms, let there be  $M$  training 2-D image samples in a facial dataset  $\{X\}$  ( $x \in R^{I_1 \times I_2}$ ). The objective of the bilateral 2-D based approach is to perform the following projection:  $y_m = U^T x_m V, m = 1, 2, \dots, M$ , where  $U \in R^{I_1 \times P_1}$  ( $P_1 < I_1$ ), and  $V \in R^{I_2 \times P_2}$  ( $P_2 < I_2$ ), are left- and right- projection matrices. Projecting each object  $x_m$  using the projection function  $y_m$  yields a feature matrix of size  $(P_1 \times P_2)$ . Then, fed these extracted features  $\{y_m\}$  are fed into a classifier for classification purposes, and the decision about similarity is considered based on a measure of distance. Inspired by previous work [51-53], an analogous model is present in chapter 4 which is called Bidirectional Two-Dimensional Neighborhood Preserving Projectio (B2DNPP) an attempt in to improve the performance of the (2D-NPP) method [50].

Different facial objects naturally have some specific structures in the form of third- or even higher-order tensors. For instance, colour images and also the sequence data of gray images are in the form of third-order tensors. Therefore in research into human face recognition, it is important to use 3-D information from column, row, and depth dimensions, including higher order data, in order to reveal the essential structures for data analysis. Linear subspace learning (LSL) algorithms are traditional dimensionality reduction techniques that represent input data as vectors or 2-D matrices and used for optimal linear mapping to a lower-dimensional space. Unfortunately, they are usually inadequate in dealing with tensorial data, and they result in losing the natural structures and correlations in the original multidimensional data [81]. Thus, multilinear subspace learning is desirable because it operates directly on the tensor objects and offers great potential in processing them. The general problem of multilinear subspace learning is formulated in the next chapter.

## 2.5 Summary

Linear subspace learning (LSL) is a popular approach in image recognition applications. It aims to reveal the important features of high-dimensional data such as facial images, in a lower dimensional space via linear projection. This chapter has presented a brief review of approaches to dimensionality reduction. Some traditional linear dimensionality reduction techniques were discussed from the perspective of face recognition, such as principal component analysis (PCA), linear discriminant analysis (LDA), locality preserving projection (LPP) and neighborhood preserving projection (NPP). The PCA and LDA effectively see only the Euclidean structure of face space, whilst LPP and NPP find an optimal projection that preserves the intrinsic structure of the data. In addition, a face subspace is obtained that best identifies the most important aspects of the manifold structure of the face. The local information of the images preserved by LPP and NPP is more significant than the global structure preserved by PCA and LDA, especially in face recognition applications. All classical methods of linear discriminant analysis used linear algorithms that operate on one-dimensional objects (vectors), which is computationally expensive. Furthermore, good performance might not result when the number of training samples is small. On the other hand, two-dimensional and the bidirectional two-dimensional linear methods require less computational time and improve the accuracy of recognition. The main difference between two-dimensional and bidirectional two-dimensional methods is that the former only perform in the row direction of the face images, while the latter work simultaneously in the row and column directions. Therefore, the number of coefficients needed bidirectional two-dimensional approaches to face representation and recognition is much smaller than in two-dimensional methods.

In fact, natural facial images are the composite consequence of many factors and modes. Thus, it is better to extract useful information from tensorial data directly rather than from vectors or matrices. Multilinear algebra thus offers a natural approach to the handling of images that are the consequence of any number of multilinear factors and to addressing the difficult problem of disentangling the constituent factors or modes.

# Chapter 3:

## Multilinear Subspace Learning, and Facial Database

---

Face images are naturally in the form of second- or higher- order tensors, and also video sequences with 2-D gray level images can be viewed as third-order tensors with column, row, and depth modes [82]. Therefore, in the most active area of biometrics research, and especially in the fields of face recognition and face detection, the use of third-order tensors has become an important research direction [82-84]. One of the key problems in data analysis in pattern recognition and computer vision is finding a suitable type of data representation [6], and it is appropriate to handle tensor representation directly rather than with vector or matrix representations. Multilinear subspace learning (MSL) approaches operate directly on the original natural tensorial data, and so the problem of tensor representation problem in object feature extraction and classification is solved [26, 45]. This chapter provides an overview of MSL techniques for the reduction of dimensionality in multidimensional data directly from their natural representations. The background of MSL is firstly introduced. Then basic multilinear algebra is reviewed and tensor distance measurement is explained. After that, the multilinear tensor-to-tensor projection is explained. Next, the problem of multilinear subspace learning is briefly described. Afterwards, some existing supervised and unsupervised MSL algorithms are reviewed, highlighting their limitations. Thereafter, a wide range of MSL applications are discussed. A review of three facial databases commonly used in the evaluation. After that, an overview of performance evaluation mechanisms is provided. Finally, the central issues of MSL are summarized.

### 3.1 Introduction

Huge multidimensional images are routinely created by various applications, made possible by the rapid advances in storage capacity and data collection methods. These natural images are the composite consequence of multiple factors and modes related to scene structure, illumination, imaging and viewpoint. Important factors in face recognition include different facial expressions, head poses, and lighting conditions.

Therefore, the learning algorithms used to extract important information from this massive amount of data are becoming crucial. Actually, two-dimensional (2-D) data are used in many research areas, including gray images in pattern recognition and computer vision [85-88], and bioinformatics research [89-91], while, three-dimensional (3-D) data are used in analysing, for example video sequences with 2-D gray images in recognition activity during human-computer interaction (HCI) and surveillance [6, 92], medical image analysis [93], and research into generic object recognition [94].

All the linear-based algorithms [14, 15, 30, 33, 68] handle tensorial data as 1-D vectors or 2-D matrices, which results in high computation costs and memory demand, and also loses the natural structure in the original tensorial data. This motivated the recent development of multilinear subspace learning methods [21, 39, 44-46, 73, 86, 95] which extract features directly from the tensorial representation rather than via vectors or matrices. The tensorial data in very large images often contains large amounts of redundancy, and important information in many cases occupies only a subspace of the original space [96]. Therefore, it is important to reduce the dimensions of the original tensorial data before any modelling process takes places [30]. Dimensionality reduction techniques extract the essential features from the data by mapping the high-dimensional data space on to a low dimensional feature space in order to remove redundant and irrelevant data while retaining as much information as possible [66], as previously explained in chapter 2. In fact, MSL has more potential than LSL to extract useful and compact feature representations [97]. Moreover, significant future impact is expected from the development of new MSL algorithms and also various problems in applications may be solved including those concerning tensorial data [98]. Therefore, in the last few years, interest has grown in the use of MSL approaches [26, 46, 81, 99-102] for the reduction of dimensionality in multidimensional facial data which are represented in their natural tensorial form instead being converted into long vectors. MSL is relevant in tensor data analysis and tensor-based computation and modelling. Therefore, the next section introduces basic multilinear algebra and the measurement of tensor distances.

### **3.2 Basis of the Multilinear Approach**

This section reviews the notation, fundamental concepts and definitions of multilinear algebra, which are necessary to use in defining MSL operations [103-105]. More

detailed surveys of tensors and tensor operations can be found in the literature [23, 25, 104-108].

### 3.2.1 Notations

This thesis, follow the conventions of notation in the multilinear algebra, pattern recognition and adaptive learning literature [103, 104]. Tensors are denoted by calligraphic uppercase letters ( $\mathcal{A}, \mathcal{B}, \dots$ ), matrices by bold uppercase letters ( $\mathbf{A}, \mathbf{B}, \dots$ ), vectors by bold lowercase letters ( $\mathbf{a}, \mathbf{b}, \dots$ ) and scalars by lowercase letters ( $a, b, \dots$ ). Since vectors are first order tensor where  $N = 1$  and matrices are second order tensor where  $N = 2$ , the generalization of vectors and matrices to a higher order can be defined as tensor with  $N > 2$ . A tensor is defined as a multidimensional matrix or n-mode matrix. The elements of tensor are denoted with indices in parentheses. Indices are denoted by lowercase letters and span the range from 1 to the uppercase letter of the index, e.g.  $n = 1, 2, \dots, N$ . In order to address part of any vector, matrix, and tensor, the symbol ":" denotes the full range of the corresponding index, and  $n_1 : n_2$  denotes indices varying from  $n_1$  to  $n_2$ .

The applications targeted in this thesis focus on real-value data only, such as gray- and colour- level facial images shown in Figure 3.1, and therefore the discussion throughout this thesis is limited to real-value vectors, matrices, and tensors.

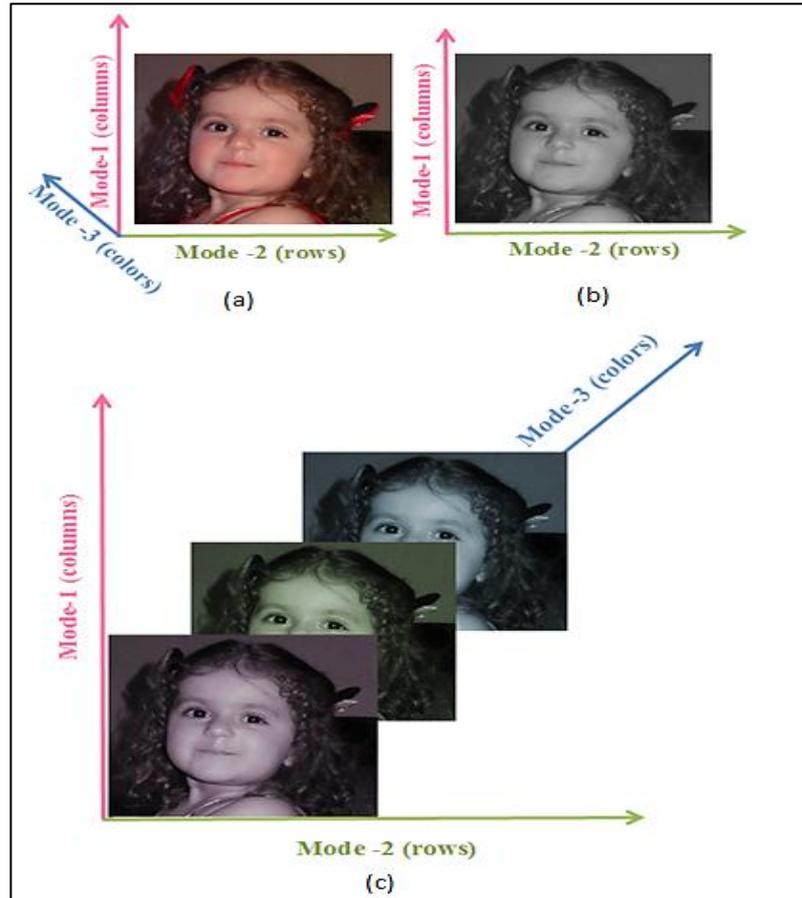


Figure 3.1: Real-world images in their natural tensor representation: (a) a colour-level image, (b) a grey-level image, and (c) 3-channels (red, green, and blue) of the colour-level image.

### 3.2.2 Multilinear algebra

Multilinear algebra is the basis of tensor data analysis, and has been studied in mathematics for several decades [23, 104, 106]. The tensor  $\mathcal{X} \in R^{I_1 \times I_2 \times \dots \times I_N}$  is from the order  $N$ , so it is described by  $N$  indices,  $i_n = 1, 2, \dots, N$  and each  $i_n$  addresses the mode- $n$  of  $\mathcal{X}$ . The  $i_n$ th mode- $n$  slice of the tensor  $\mathcal{X}$  is an  $(N-1)$ th-order tensor obtained by fixing the mode- $n$  index of  $\mathcal{X}$  as  $\mathcal{X}(:, \dots, :, i_n, :, \dots, :)$ . The mode- $n$  vectors of tensor  $\mathcal{X}$  are defined as the  $I_n$ -dimensional vectors obtained from  $\mathcal{X}$  by varying the index  $i_n$  and keeping the other indices fixed. A rank-1 tensor  $\mathcal{X}$  equals to the outer product of  $N$  vectors as follows:

$$\mathcal{X} = \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ \dots \circ \mathbf{u}^{(N)}, \quad (3.1)$$

this means that:

$$\mathcal{X}(i_1, i_2, \dots, i_N) = \mathbf{u}^{(1)}(i_1) \cdot \mathbf{u}^{(2)}(i_2) \cdot \dots \cdot \mathbf{u}^{(N)}(i_N), \quad (3.2)$$

for all values of indices. Unfolding the tensor  $\mathcal{X}$  over the mode- $n$  is denoted as  $\mathcal{X}^{(n)} \in \mathbf{R}^{I_n \times (I_1 \times \dots \times I_{n-1} \times I_{n+1} \times \dots \times I_N)}$ , where the column vectors of  $\mathcal{X}^{(n)}$  are the mode- $n$  vectors of  $\mathcal{X}$ . Figure 3.2(b), (c), and (d) respectively illustrate the mode-1, mode-2, and mode-3 slices of the tensor  $\mathcal{X}$  in Figure 3.2(a). Figure 3.3(b), (c), and (d) shows the mode-1, mode-2, and mode-3 unfolding respectively of the tensor  $\mathcal{X}$  in Figure 3.3(a).

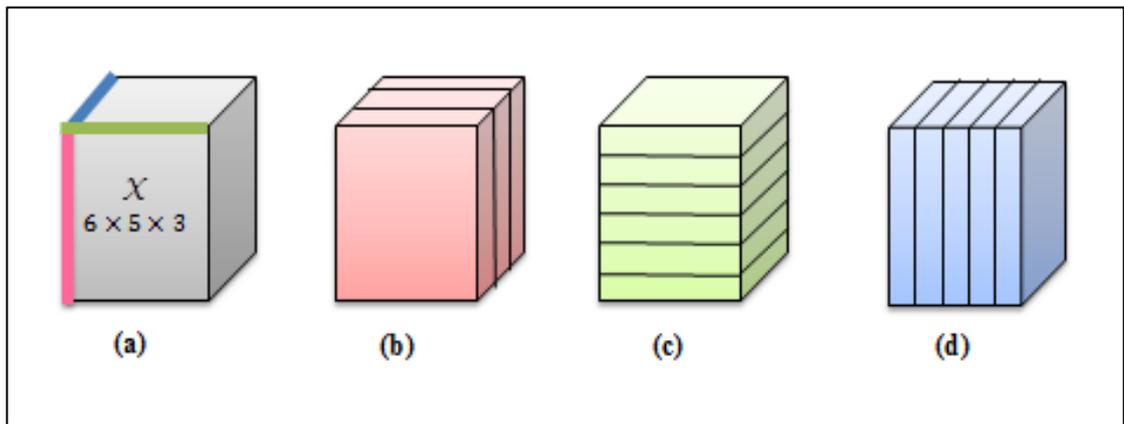


Figure 3.2: Example of mode- $n$  vectors: (a) a tensor  $\mathcal{X} \in \mathbf{R}^{6 \times 5 \times 3}$ , (b) the mode-1 slices, (c) the mode-2 slices, and (d) the mode-3 slices.

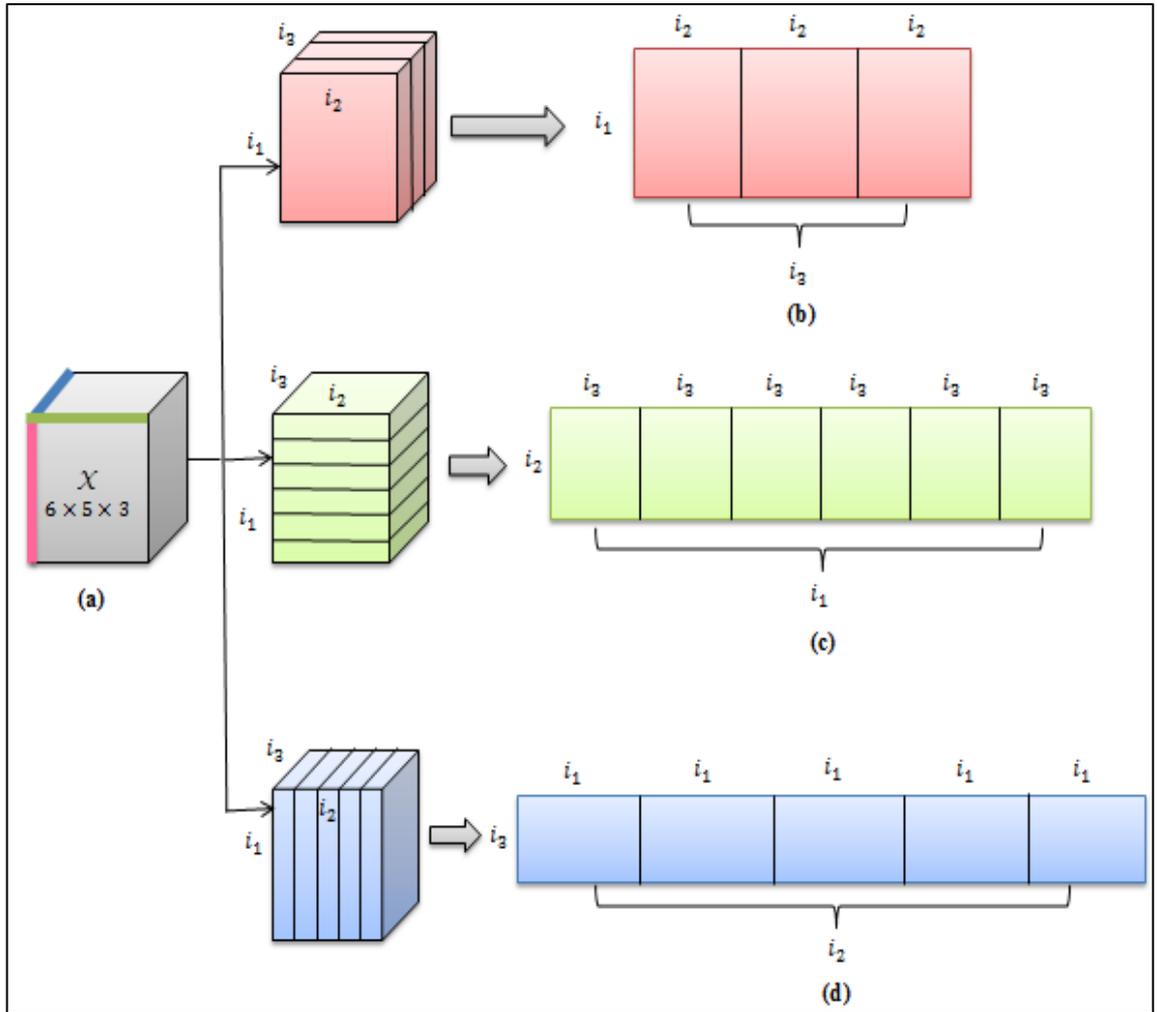


Figure 3.3: Example of unfolding the third-order tensor to the three mode- $n$  matrices: (a) a tensor  $\mathcal{X} \in R^{6 \times 5 \times 3}$ , (b) mode-1 matrix (columns), (c) mode-2 matrix (rows), and (d) mode-3 matrix (colours).

Here, some essential definitions in the multilinear approach are recalled that will be useful in what follows in this thesis. These definitions have been explained in detail in earlier research such as [104, 106, 109].

Definition 1: In tensor terms, the tensor  $\mathcal{X}$  can be written as the product:

$$\mathcal{X} = \mathbf{y}_{\times_1} \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times \dots \times_N \mathbf{U}^{(N)}, \quad (3.3)$$

where

$$\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}^{(1)T} \times_2 \mathbf{U}^{(2)T} \times \dots \times_N \mathbf{U}^{(N)T} . \quad (3.4)$$

The size of tensor  $\mathcal{Y}$  is  $(I_1 \times I_2 \times \dots \times I_N)$ , where  $\mathbf{U}^{(n)} = [\mathbf{u}_1^{(n)} \mathbf{u}_2^{(n)} \dots \mathbf{u}_{I_n}^{(n)}]$ ,  $n = 1, 2, \dots, N$  is a matrix of size  $(I_n \times I_n)$  that contains orthonormal column vectors  $\|\mathcal{X}\|_F^2 = \|\mathcal{Y}\|_F^2$  [60]. The mode- $n$  decomposition matrix obtained by unfolding the tensor  $\mathcal{Y}$ , can be represented in a matrix format [46, 110] as:

$$\mathbf{Y}^{(n)} = \mathbf{U}^{(n)T} \tilde{\mathbf{X}}^{(n)} . \quad (3.5)$$

where

$$\tilde{\mathbf{X}}^{(n)} = \mathbf{X}^{(n)} (\mathbf{U}^{n+1} \otimes \dots \otimes \mathbf{U}^N \otimes \mathbf{U}^1 \otimes \dots \otimes \mathbf{U}^{n-1}) \quad (3.6)$$

Here  $\otimes$  denotes the Kronecker product, while  $\tilde{\mathbf{X}}^{(n)}$  in terms of the tensor is defined as:

$$\tilde{\mathbf{X}}^{(n)} = \mathcal{X} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times \dots \times_{n-1} \mathbf{U}^{(n-1)} \times_{n+1} \mathbf{U}^{(n+1)} \times \dots \times_N \mathbf{U}^{(N)} . \quad (3.7)$$

The unfolding matrix of tensor  $\mathcal{X}$  can then be written as:

$$\mathbf{Y}^{(n)} = \mathbf{U}^{(n)T} \mathcal{X} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times \dots \times_{n-1} \mathbf{U}^{(n-1)} \times_{n+1} \mathbf{U}^{(n+1)} \times \dots \times_N \mathbf{U}^{(N)} \quad (3.8)$$

In short, it can be written as

$$\mathbf{Y}^{(n)} = \mathbf{U}^{(n)T} \mathcal{X} \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} \quad (3.9)$$

The dimension of the unfolding matrix  $\mathbf{Y}^{(n)}$  is therefore given by  $(I_n \times (I_1 \times I_2 \times \dots \times I_{n-1} \times I_{n+1} \times \dots \times I_N))$ . A more detailed survey of tensors and tensor operations has been published in [111].

Definition 2: The mode- $n$  product of a tensor  $\mathcal{X} \in R^{I_1 \times I_2 \times \dots \times I_n \times \dots \times I_N}$  by the matrix  $\mathbf{U} \in R^{I_n \times P_n}$ , is denoted as  $\mathcal{B} = (\mathcal{X} \times_n \mathbf{U}) \in R^{I_1 \times \dots \times I_{n-1} \times P_n \times I_{n+1} \times \dots \times I_N}$  [106] in terms of tensors, while in matrix terms it is denoted as  $\mathbf{B}^{(n)} = \mathbf{U}^T \mathbf{X}^{(n)}$ . Entries of the tensor  $\mathcal{B}$  can be denoted as:

$$(\mathcal{X} \times_n \mathbf{U})(i_1, \dots, i_{n-1}, P_n, i_{n+1}, \dots, i_N) = \sum_{i_n} \mathcal{X}(i_1, \dots, i_n) \cdot \mathbf{U}(P_n, i_n). \quad (3.10)$$

Definition 3: The scalar product of two  $\mathcal{X}, \mathcal{Z} \in R^{I_1 \times I_2 \times \dots \times I_N}$  tensors of the same order and dimensions is given by:

$$\langle \mathcal{X}, \mathcal{Z} \rangle = \sum_{i_1} \sum_{i_2} \dots \sum_{i_N} \mathcal{X}(i_1, i_2, \dots, i_N) \cdot \mathcal{Z}(i_1, i_2, \dots, i_N) \quad (3.11)$$

Definition 4: The Forbenius norm of tensor  $\mathcal{X}$  is defined as:

$$\|\mathcal{X}\|_F = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle} = \left( \sum_{i_1} \sum_{i_2} \dots \sum_{i_N} \mathcal{X}^2(i_1, i_2, \dots, i_N) \right)^{1/2}. \quad (3.12)$$

Definition 5: The distance between tensors  $\mathcal{X}$  and  $\mathcal{Z}$  can be measured by the Frobenius norm:

$$dist(\mathcal{X}, \mathcal{Z}) = \|\mathcal{X} - \mathcal{Z}\|_F. \quad (3.13)$$

The tensor-based measure is equivalent to a distance measure of corresponding vector representations, as previously proven [112]. Let  $vec(\mathcal{X})$  be the vector representation of the tensor  $\mathcal{X}$ , then:

$$dist(\mathcal{X}, \mathcal{Z}) = \|vec(\mathcal{X}) - vec(\mathcal{Z})\|_2. \quad (3.14)$$

This means that the distance between two tensors as defined in Equation 3.13 equals the Euclidean distance between their vectorized representations.

### 3.3 Multilinear Projection

The multilinear subspace can be defined as the multilinear projection that maps the tensorial data from multilinear space to a lower-dimensional space [22] as shown in Figure 3.4. There are three basic types of multilinear projection based on the input and output of the projection. The tensor-to-scalar projection is also known as elementary multilinear projection (EMP). The tensor-to-vector projection is referred to as the rank-one projections in some studies [37, 38], and the tensor-to-tensor projection [22, 106], the type considered in this thesis.

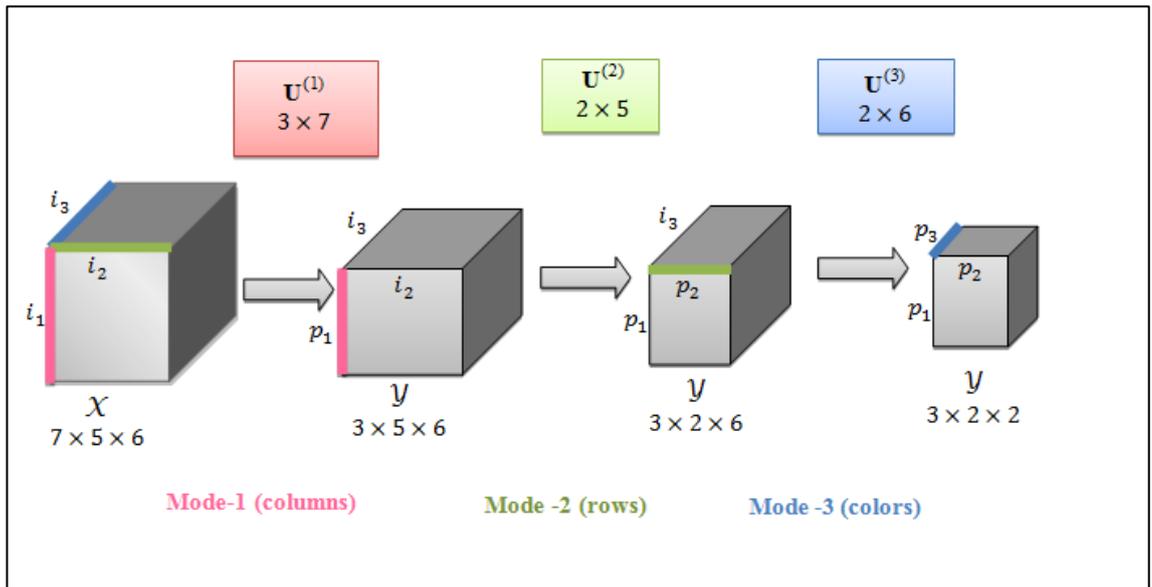


Figure 3.4: Projection of a 3-order tensor in all modes

An  $N$ -order tensor  $\mathcal{X}$  in the space  $R^{I_1 \times I_2 \times \dots \times I_n \times \dots \times I_N}$  can be projected to a new tensor  $\mathcal{Y}$  of the same order in a lower-dimensional tensor space  $R^{P_1 \times P_2 \times \dots \times P_n \times \dots \times P_N}$ , where  $P_n < I_n$ , for all  $n=1,2,\dots,N$ . This projection can be conducted in  $N$  steps according to the order of the tensor. In each step, each mode- $n$  vector of the original tensor is projected to a lower dimension  $P_n$  using  $N$  projection matrices  $\mathbf{U}^{(n)} \in R^{I_n \times P_n}$  for all  $n=1,2,\dots,N$ . Figure 3.5 explains the projection of a tensor in mode-1 using a mode-1 projection matrix  $\mathbf{U}^{(1)}$ .

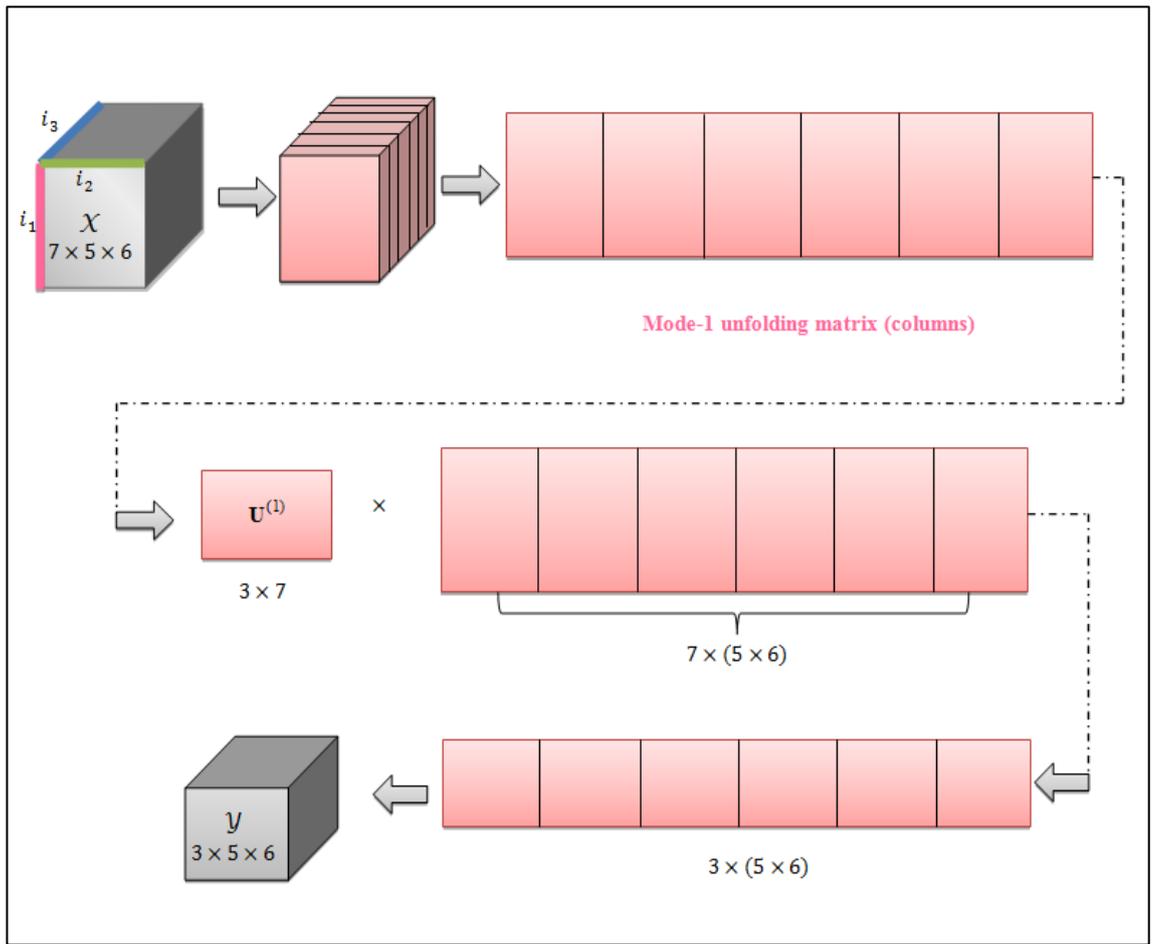


Figure 3.5: Projection of a 3-order tensor in mode-1

### 3.4 Multilinear Subspace Learning Approach

This section defines the general multilinear subspace learning framework, and illustrates the problem with MSL. In addition, it formulates a typical multilinear subspace learning algorithm.

#### 3.4.1 MSL problem definition

Multilinear subspace learning solves a multilinear projection with some optimality criteria, so that it is an extension of the linear subspace learning approach (LSL). The problem involved in the MSL approach based on tensor-to-tensor projection can be mathematically formulated in a similar way to that of the definition of LSL given in Section 2.2.1 as follows:

Let  $\{\mathcal{X}\}$  be a facial dataset of  $M$  training tensorial samples, where each object  $\mathcal{X}_m$  is an  $N$ -order tensor ( $\mathcal{X}_m \in R^{I_1 \times I_2 \times \dots \times I_N}, m=1,2,\dots,M$ ). The objective of tensor-to-tensor projection in multilinear subspace learning is to find multilinear projection matrices  $\mathbf{U}^{(n)} \in R^{I_n \times P_n}$  for all  $n=1,2,\dots,N$ . These transformation matrices are used to transform, or map each tensor from the original space  $R^{I_1} \otimes R^{I_2} \otimes \dots \otimes R^{I_N}$  into a space of lower dimension order  $R^{P_1} \otimes R^{P_2} \otimes \dots \otimes R^{P_N}$  with  $P_n < I_n$  for all  $n=1,2,\dots,N$ , using the mapping function  $\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}^{(1)T} \times_2 \mathbf{U}^{(2)T} \times \dots \times_N \mathbf{U}^{(N)T}$  such that the new projected tensors satisfy an optimality criterion. This problem is solved using an iterative alternating projection method. Then, in the classification step, the featuring  $\{\mathcal{Y}_m\}$  are fed into a classifier such that the nearest neighbor classifier, and similarity is usually computed based on some distance measure based on the Frobenius norm shown in Equation 3.14. The general MSL problem can be written mathematically as follows:

$$\{\mathbf{U}^{(n)}, n=1,2,\dots,N\} = \arg \max_{\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \dots, \mathbf{U}^{(N)}} \varphi(\mathcal{Y}), \quad (3.15)$$

or

$$\{\mathbf{U}^{(n)}, n=1,2,\dots,N\} = \arg \min_{\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \dots, \mathbf{U}^{(N)}} \varphi(\mathcal{Y}), \quad (3.16)$$

where  $\varphi(\cdot)$  denotes a criterion function to be maximized or minimized.

### 3.4.2 General multilinear approach

One of the key differences between linear and multilinear subspace learning is the method of solving the optimization problem. The linear subspace approach based on vector-to-vector projection has a closed-form solution, whereas, tensor-based (tensor-to-tensor, tensor-to-vector and tensor-to-scalar) projections in the multi-linear subspace approach have  $N$  set of parameters determined by the order of the tensor space to be solved; one for each mode. The solution for one set usually depends on the other sets, which makes their simultaneous estimation very difficult or even impossible. Therefore, an alternating projection using the alternating least square (ALS) method [113, 114] is often adopted in order to solve the tensor-based projections for the optimization problem of all modes. This solves a set of parameters in one mode at a time. The ALS method was first implemented in 1970 to solve the parameter estimation problem in 3-

dimensional factor analysis [115]. The ALS aims to reduce the optimization problem into smaller sub-problems that can be solved using methods determined for use in the linear case. Therefore, in each mode, the parameters are estimated separately and are conditional on the values of other parameters in other modes. The optimal base parameters for each mode are obtained iteratively by fixing the base parameters on the other modes and cycle for the remaining variables. Due to this iterative solution, various problems should be addressed, such as initialization and termination as well as determining the desired dimensionality of the  $P_1, P_2, \dots, P_N$  of the subspace. Figure 3.6 illustrates a typical and general multilinear subspace learning pseudo code.

**Input:** Original tensor samples set  
**Output:** The multi-linear projections that minimizes/ maximizes an optimal criterion in the projection space.  
**Algorithm:**  
    **Step 1 (Initialization)**  
        Initialize the projection matrices.  
    **Step 2 (Local optimization):**  
        Repeat:  
            For t=1: total number of iterations  
                For n= 1: number of directions  
                    Solve the multi-linear projection for each mode- $n$   
                End for (n)  
        End for (t)

Figure 3.6: Typical multilinear subspace learning algorithm pseudo-code.

### 3.5 Relevant Multilinear Algorithms in previous Researches

As mentioned previously in Section 2.6, linear subspace learning methods such as PCA, LDA, LPP and NPP used for face recognition applications need to reshape the data samples into long vectors in a very high-dimensional space. This requires a large number of parameters to be estimated, and results in high computational costs and memory demands. In addition, the natural structure and correlation among the natural tensorial data forms is lost, from which more compact or useful representations could be obtained [39, 40]. This inspired the recent development of multilinear subspace learning algorithms, which operate directly on the tensorial representations rather than their vectorial forms. Therefore, this section reviews three multilinear subspace learning

methods that have extended the linear subspace learning methods PCA, LDA, LPP into its multilinear subspace learning forms.

Table 3.1: Overview of existing linear and multilinear subspace learning approaches.

The pink shaded empty boxes indicate a lack of relevant studies so far.

Methods	Dimensionality			
	1-D	2-D	Bidirectional 2-D	MD (Tensor)
PCA	PCA	2DPCA	(2D)2DPCA	MPCA
LDA	LDA	2DLDA	(2D)2DLDA	DATER
LPP	LPP	2DLPP	B2DLPP	ND-TSNE
NPP	NPP	2DNPP		

Table 3.1 gives an overview of some of the existing linear subspace learning algorithm that introduced in Chapter 2, along with the corresponding multilinear subspace learning algorithms based on tensor-to-tensor projection in the multilinear subspace learning framework which are discussed below:

### 3.5.1 Multilinear principal component analysis (MPCA)

A multilinear extension of the PCA method, called the multilinear principal component analysis (MPCA) of tensor objects was proposed in 2008 [46]. This algorithm solves the optimization problem through an iterative procedure by decomposing the problem into  $N$  linear problems. It maximizes the total tensor scatter as follows:

$$\Psi \mathcal{Y} = \sum_{m=1}^M \left\| \mathbf{y}_m - \bar{\mathbf{y}} \right\|_F^2, \quad (3.17)$$

where  $\bar{\mathbf{y}}$  is the mean sample defined as follows:

$$\bar{\mathbf{y}} = \frac{1}{M} \sum_{m=1}^M \mathbf{y}_m. \quad (3.18)$$

Since it decomposes the optimization problem into smaller sub-problems, it solves the following eigenvalue problem in each sub-problem:

Given all the projection matrices  $\mathbf{U}^{(k)} \in R^{I_k \times P_k}$   $P_n < I_n$  for  $k = 1, 2, \dots, N, k \neq n$ , the matrix  $\mathbf{U}^{(n)} \in R^{I_n \times P_n}$  that maximizes  $\Psi \mathcal{Y}$  consists of  $P_n$  eigenvectors corresponding to the largest  $P_n$  eigenvalues, and it can be found as follows:

$$\boldsymbol{\varphi}^{(n)} = \sum_{m=1}^M \left( \mathbf{X}_m^{(n)} - \bar{\mathbf{X}}^{(n)} \right) \cdot \tilde{\mathbf{U}}^{(\varphi_{(n)})} \cdot \tilde{\mathbf{U}}^{(\varphi_{(n)})T} \cdot \left( \mathbf{X}_m^{(n)} - \bar{\mathbf{X}}^{(n)} \right)^T, \quad (3.19)$$

where,

$$\tilde{\mathbf{U}}^{(\varphi_{(n)})} = \left( \tilde{\mathbf{U}}^{(1)} \otimes \tilde{\mathbf{U}}^{(2)} \otimes \dots \otimes \tilde{\mathbf{U}}^{(n-1)} \otimes \tilde{\mathbf{U}}^{(n+1)} \otimes \dots \otimes \tilde{\mathbf{U}}^{(N)} \right). \quad (3.20)$$

The produced projection matrices  $\mathbf{U}^{(n)} \in R^{I_n \times P_n}$ , where  $P_n < I_n$  for all  $n=1,2,\dots,N$  could be written as  $\prod_{n=1}^N P_n$  EigenTensor:  $\tilde{\mathbf{U}}_{p_1 p_2 \dots p_N} = \tilde{u}_{p_1}^{(1)} \circ \tilde{u}_{p_2}^{(2)} \circ \dots \circ \tilde{u}_{p_N}^{(N)}$ , where each  $\tilde{u}_{p_n}^{(n)}$  denotes the  $p_n$ th column of  $\tilde{\mathbf{U}}^{(n)}$ . However, not all of the EigenTensors are useful for accurate recognition. Therefore, they may be selected according to their class discrimination as follows:

$$\Gamma_{p_1 p_2 \dots p_N} = \frac{\sum_{c=1}^C M_c \cdot \left[ \bar{\mathbf{y}}_c(p_1, p_2, \dots, p_N) - \bar{\mathbf{y}}(p_1, p_2, \dots, p_N) \right]^2}{\sum_{m=1}^M \left[ \mathbf{y}_m(p_1, p_2, \dots, p_N) - \bar{\mathbf{y}}_{c_m}(p_1, p_2, \dots, p_N) \right]^2}, \quad (3.21)$$

where  $C$  is the number of classes,  $M_c$  is the number of samples for each class ( $c$ ),  $\mathbf{y}_m$  is the feature tensor object obtained for the input tensor  $\mathcal{X}_m$  in the low projected subspace, and  $\bar{\mathbf{y}}_c$  is the class mean feature tensor that can be calculated as follows:

$$\bar{\mathbf{y}}_c = \frac{1}{M} \sum_{m, c_m=c} \mathbf{y}_m. \quad (3.22)$$

Then, the elements of each projected tensor  $\mathbf{y}_m$  are reshaped into the feature vector  $y_m$  and rearranged in descending order according to  $\Gamma_{p_1 p_2 \dots p_N}$ , and only the first  $P$  largest elements of each  $y_m$  are then kept. In classification, the new feature vectors are used alone or combined with LDA (MPCA+LDA) [116] as an input to the nearest neighbor classifier.

This method has been applied in both face and gait recognition. But it appears to be sensitive to the setting of the  $P$  value, which is used to further reduce the dimensionality of the projected tensors. In addition, it can be performed only in an unsupervised way, so it does not require any additional information about class labels during the calculation of total tensor scatter. Moreover, the MPCA only preserves the global structure of the data samples and ignores important local information.

### 3.5.2 Discriminant analysis with tensor representation (DATER)

In 2007, Yan et al [26] developed a multilinear extension of the LDA method called discriminant analysis with tensor representation (DATER). This aims to implement the discriminant analysis directly on the natural tensorial data. As with MPCA, DATER cannot solve the optimization problem deterministically for the tensor-to-tensor projection. Thus, an iterative alternating projection procedure is followed in the same way as in MPCA and any other typical multilinear subspace approach.  $N$  projection matrices  $\mathbf{U}^{(n)} \in R^{I_n \times P_n}$  are obtained for all  $n = 1, 2, \dots, N$ , which used later to project each  $N$ -order input tensor  $\mathcal{X}_m \in R^{I_1 \times I_2 \times \dots \times I_N}$  into a smaller tensor  $\mathcal{Y}_m \in R^{P_1 \times P_2 \times \dots \times P_N}$ , where  $P_n < I_n$  for all  $n = 1, 2, \dots, N$ . This method formulates the objective criterion for tensor discrimination and maximizes the ratio of between-class scatter  $S_B$  to within-class scatter  $S_W$ , where:

$$S_B = \sum_{c=1}^C M_c \left\| \overline{\mathcal{Y}}_c - \overline{\mathcal{Y}} \right\|_F^2, \quad (3.23)$$

and,

$$S_W = \sum_{m=1}^M \left\| \mathcal{Y}_m - \overline{\mathcal{Y}}_{c_m} \right\|_F^2, \quad (3.24)$$

where  $C$  is the number of classes,  $M_c$  is the number of samples for each class  $c$ ,  $c_m$  is the class label for the  $m$ th object sample,  $\mathcal{Y}_m$  is the extracted feature tensor sample of the input tensor  $\mathcal{X}_m$ ,  $\overline{\mathcal{Y}}$  is the mean tensor of all the obtained tensors, and  $\overline{\mathcal{Y}}_c$  is the class mean feature tensor that can be found as Equation 3.22. The transform is then determined by deriving the eigenvectors from the resulting ratio of the between-class scatter to the within-class scatter. After the tensorial input images from the original space has been transformed into lower-dimension space via the learned subspaces, the nearest neighbor method is used as the classifier in the final classification step.

The DATER method has been applied successfully in face recognition applications [26, 77, 102], whereas the application of DATER to gait recognition [87] appears to be sensitive to parameter settings. But, similar to the MPCA, the DATER preserves only the global structure of the tensorial data samples at the cost of losing significant information concerning to the local geometry of the neighbors. Moreover, this algorithm is implemented only in a supervised method, which requires additional information about the class labels.

### 3.5.3 Multilinear tensor supervised neighborhood embedding (ND-TSNE)

An  $N$ -dimensional tensor supervised neighborhood embedding (ND-TSNE) approach as introduced in 2010 for discriminant feature representation [44]. This is the multilinear form of the LPP method, which avoids the need to vectorize tensorial data samples and aims to preserve the neighborhood structure of tensor feature space. Similar to the MPCA and DATER, a deterministic solution does not exist, therefore the ND-TSNE iteratively operates an alternative least square projection approach. It is used to solve the tensor-to-tensor projection matrices  $\mathbf{U}^{(n)} \in R^{I_n \times P_n}$ ,  $P_n < I_n$  for all  $n = 1, 2, \dots, N$ , which transforms an  $N$ -dimensional input tensor  $\mathcal{X}_m \in R^{I_1 \times I_2 \times \dots \times I_N}$  into  $\mathcal{Y}_m \in R^{P_1 \times P_2 \times \dots \times P_N}$ . This is similar to the MPCA and DATER algorithms, but with a different objective criterion. This algorithm builds a nearest neighbor graph to model the local geometrical structure and labels information from the data using a weight matrix formulated according to Equation 2.21 which is used in the LPP method as explained in Section 2.5. Then the graph structure is obtained by solving the following objective functions:

$$\min_{\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_N} \sum_{ij} \left\| \mathcal{X}_i \times_1 \mathbf{U}^{(1)T} \times \dots \times_N \mathbf{U}^{(N)T} - \mathcal{X}_j \times_1 \mathbf{U}^{(1)T} \times \dots \times_N \mathbf{U}^{(N)T} \right\| \mathbf{W}_{ij} \quad 3.25)$$

After extracting the projected TSNE tensor object  $\mathcal{Y}_m$ , the extracted feature vectors are used for classification.

This approach has been applied to view-based object recognition and classification. Like the LPP, the ND-TSNE is sensitive to the width of Gaussian envelope selected. Moreover, it can be performed only in a supervised method, in the same way as DATER.

## 3.6 Facial Databases

In this thesis, three widely used public benchmark facial databases are used to evaluate the effectiveness of proposed methods. These are the AT&T Database of Faces (ORL) [117], the Purdue AR Face database [118] and the Facial Recognition Technology (FERET) database [119, 120].

### 3.6.1 The ORL database

The ORL database [117] of face images is provided by the AT&T Laboratories in Cambridge. It contains some variations in illumination, facial expression (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). These images include some tilting and rotation of the face by up to 20 degrees. The ORL database consists of 400 images relating to 40 subjects (10 images for each subject). All images were normalized to a resolution of 112 x 92 pixels with (256) gray levels. Some of the ORL face images are shown in Figure 3.7.



Figure 3.7: Facial samples from four subjects within the ORL database

### 3.6.2 The AR database

Martinez and Benavente [118] created the AR database, which contains over 4000 colour images of 135 individuals (76 males and 59 females) incorporating varied facial expressions, illumination conditions and occlusions. The sample for each subject consists of 26 images that were taken during two different sessions. The first session contains 13 pictures, named from 01 to 13, and include a neutral expression (01), smile (02), anger (03), yawning(04), different lighting conditions (05–07) and different occlusions under different lighting that cover about 30% of the images for sunglasses (08–10) and a scarf (11–13). Examples of these pictures are shown in Figure 3.8(a-m), and the second session exactly duplicates the first session but two weeks later.

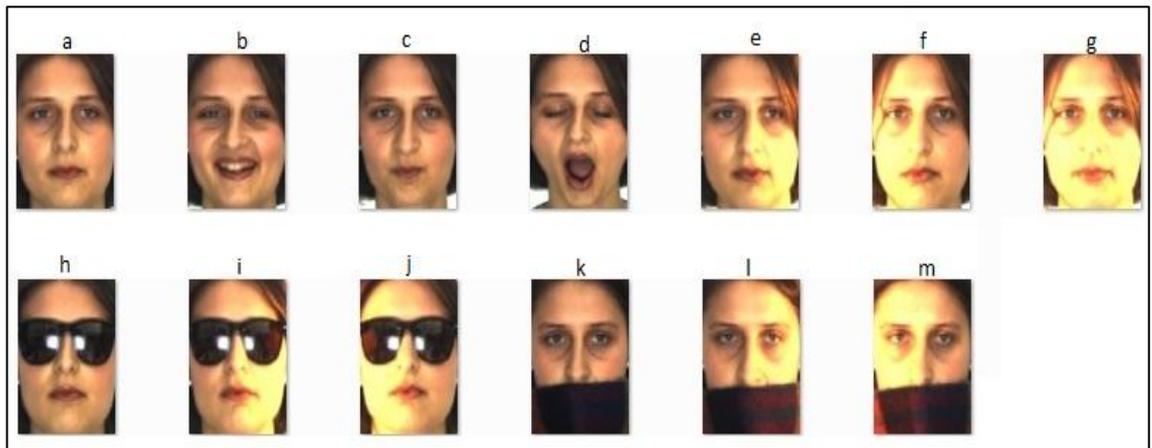


Figure 3.8: AR colour dataset sample with variations in facial expressions, lighting conditions and occlusions.

### 3.6.3 The FERET database

The FERET database [119, 120] is the most widely adopted colour image benchmark for the evaluation of face recognition methods. It was constructed by the FERET programme to support government experiments and the evaluation of face recognition algorithms. This database was collected between 1993 and 1997 in 15 sessions, and includes 14,126 images from 1199 individuals, covering a wide range of variations in viewpoint, illumination, facial expression, race and age. This database contains some images of the same person taken at different times in one or more years, resulting in various changes related to facial expression, illumination, scale, and the pose of the face. This is very important in evaluating the robustness of face recognition approaches. Therefore, it has become one of the most important, practical and standard facial database for evaluating facial recognition algorithms [121]. The selection of a FERET subset in the experimental evaluation in this thesis consists of subjects that have at least 11 images in the database with different poses. Figure 3.9 provides a sample of facial images from one subject within the FERET subset. More details of and the terminology used in this database have been published elsewhere [119, 120].

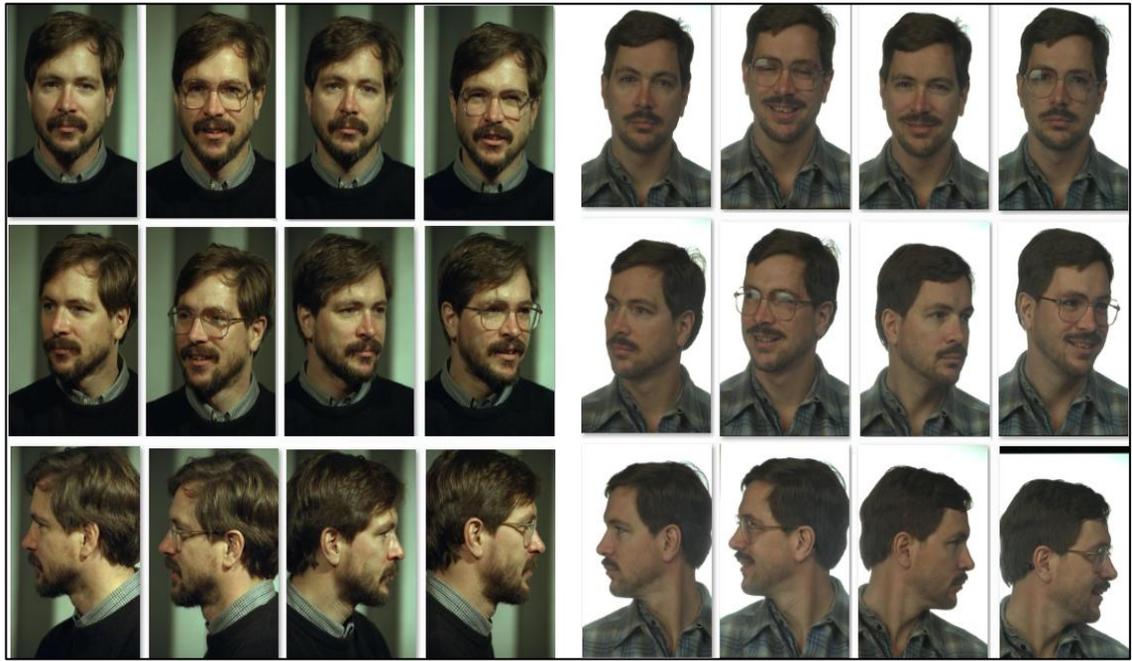


Figure 3.9: Twenty four facial samples of one subject from the FERET subset.

### 3.6.4 Dataset configuration

In a typical face recognition scenario, there are usually three types of facial dataset:

1. Gallery set: this contains a set of image samples from a number of subjects with known identities who are enrolled in the system to provide training data.
2. Probe set: this contains a set of samples to be tested from subjects with unknown identities to be classified and identified by comparing their features with those in the gallery images.
3. Outer set: this is a secondary test set consisting of images of some faces from the database that are different from those used in the gallery and probe sets. It is used to determine the false acceptance rate (FAR) and false rejection rate (FRR), especially for recognition tasks in biometric verification.

### 3.7 Pre-processing of Facial Images

The selection of a particular set of pre-processing steps differs according to the application concerned. Since this thesis focus on face recognition, all face images used from the three databases ORL, AR and FERET are manually aligned by fixing the locations of the eyes. So, firstly, all faces are rotated so that the centres of the eyes are placed on specific pixels. Then, the region of the image where the face is located is

cropped and only this area is used in the process of face recognition. The face is cropped from the entire image manually depending on the coordinates of the left and right eye and also the mouth. Finally, all the new images should have the same dimensions. Therefore, after cropping the face from the original image, the new images are normalized to a standard size of 64 x 64 pixels. In this thesis, both gray- and colour-level facial images are considered. The pre-processing procedure is shown in Figure 3.10.

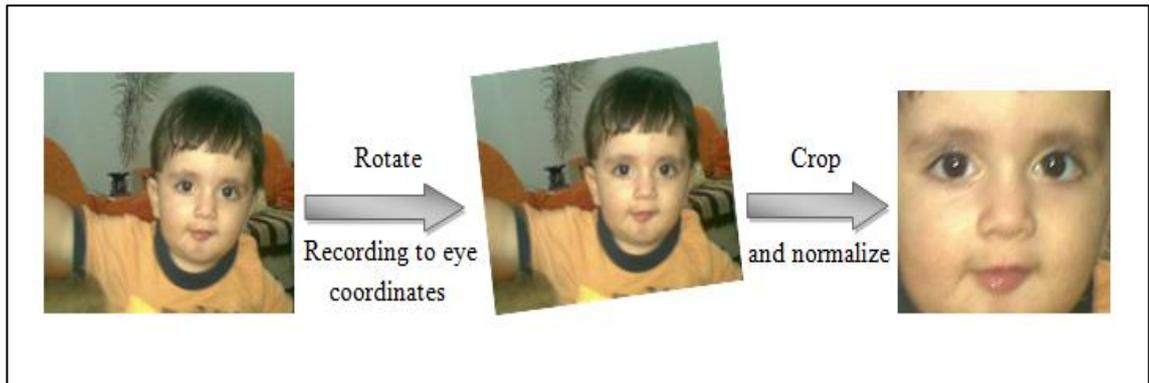


Figure 3.10: The pre-processing procedure for face images.

### 3.8 Performance Evaluation of Facial Recognition

Most biometric face recognition algorithms depend upon two types of tasks to identify the face images of persons from a database, which are defined here as follows:

1. Identification. This involves the process of comparing the person’s biometric data (the query sample) of an unknown individual against all persons (samples) in the database. The output is the identity or a list of possible identity list of the input sample. The performance of identification is measured either using the correct recognition rate (CRR), or by the cumulative match characteristic (CMC) [120].
2. Verification. This compares the biometric data (query sample) against only the entries that correspond to the users claimed identity in the database. This means that the person should provide an identity and the system either accepts or rejects according to successful or unsuccessful verification. The performance of verification can be measured by the receiver operator characteristic (ROC) which

represents the FAR as a function of the FRR [122], or by the detection error trade-off (DET) [123].

This research focuses on both of these evaluation tasks of identification and verification. It considers two kinds of face recognition problems, which are stated as follows:

1. The face recognition problem that measures the performance according to identification rate. Let A be a gallery database that consist of face images from a number of known subjects. The essential objective of the facial recognition system is to identify the identity of unknown probe images and to determine their corresponding gallery subjects. In this thesis, the performance of the system is measured using the CRR.
2. The face recognition problem that measures performance according to the verification rate. Let A and B be two databases that contain number of subjects, and in each there are a number of images for known persons. Let G is a gallery database that consists of face images from a number of known subjects from the database A and P is the probe set that consists of some unknown testing data images from the same database A, while O is the outer test image set that consist of face images from dataset B. The objective of the facial recognition system is to determine the identity of the unknown probe and the outer images and to determine their corresponding gallery subjects. In addition, the false acceptance rate (FAR) and false rejection rate (FRR) are tested of the system. The performance of the system in this thesis is also measured by the ROC.

### **3.9 Summary**

Multilinear subspace learning (MLS) approaches have the ability to handle massive amount of data and to reduce the dimensionality of multidimensional data directly from their natural tensorial representations rather than having to be converted into vectors. they also provide an effective way to extract useful and compact feature representations which are better than those of LSL methods. Moreover, it preserves the natural structure of multidimensional data and fewer parameters need to be estimated than LSL approach. This chapter has presented a brief review and the background of the multilinear subspace framework. It covered the general fundamentals of the approaches including the basic multilinear notation and algebra. It then explained the tensor-to-tensor projection, problem definition and the typical approach used in MSL. Moreover,

it reviewed some multilinear subspace learning methodologies from previous studies, and outlined their drawbacks as well. Also, this chapter described the three databases to be used in this thesis, in addition to explaining the configuration of the dataset and the pre-processing of facial images. Finally, it discussed the problem of performance evaluation of facial recognition algorithms.

There are still methods yet to be explored in both bidirectional two-dimensional and multilinear subspace learning, as explained in Table 3.1. therefore, inspired by the work in described [26, 44, 46, 51-53], chapter 4 presented an analogous model called bidirectional two-dimensional neighborhood preserving projection (B2DNPP), the aim of which is to improve the performance of the 2D-NPP method [36]. In addition, in chapter 5 a novel biometric face recognition approach is developed based on a multilinear neighborhood preserving projection algorithm (MNPP). This is used to overcome the limitations of existing MSL algorithms and to enhance the performance of B2DNPP face recognition and to naturally extend the standard 1-D NPP to the multilinear form.

# **Chapter 4:**

## **Face Recognition Approach Based on Bidirectional Two-Dimensional Neighborhood Preserving Projection and Curvelet Transform**

---

The wavelet transform is a multi-resolution tool that is widely used in the field of face recognition and applications in other areas such as pattern recognition, image processing and computer vision. Recently, a number of other multi-resolution tools have been developed, such as contourlet [124], ridgelet [125] and curvelet [126] transforms. The curvelet transform improves directional elements as well as having the ability to effectively represent curved edge. Therefore, the curvelet transform may have the capability to extract better information concerning curves and edges from human facial images compared to other multi-resolution tools.

A new method of bidirectional two-dimensional neighborhood preserving projection (B2DNPP) for human facial recognition is presented in this chapter, which uses only one weight matrix to extract two projection matrices. Unlike 2DNPP [50], which performs compression only in the column direction, B2DNPP works in both row- and column- directions to preserve spatial information about the relationship between rows and columns in the images. To further improve the performance of B2DNPP, a new facial recognition approach is proposed which is based on bidirectional two-dimensional neighborhood preserving projection (C-B2DNPP). This involves the curvelet decomposition of human facial images, where the curvelet sub-bands that demonstrate high variation are selected as initial image features. The B2DNPP algorithm is applied to preserve the local structure of the data manifold and generate distinctive feature vectors. An extreme learning machine (ELM) classifier is used for classification and this significantly improved the classification rate. Extensive

experiments have been performed to evaluate the performance of B2DNPE and C-B2DNPP with three benchmark human facial databases, ORL, AR, and FERET. It is proved that C-B2DNPP outperforms 2DNPP and all existing bidirectional 2-D methods with various databases with respect to recognition rate, computational time, and the number of selected eigenvectors. The proposed method addresses the pink shaded empty box in Table 3.1 (page 53) under bidirectional two-dimensional projection.

This chapter is organized as follows. Section 1 describes relevant previous work. Section 2 gives a brief introduction to the idea of the curvelet transform, followed by an overview of existing face recognition techniques based on the use of multi-resolution tools and different feature extraction methods. Section 3 outlines two-dimensional neighborhood preserving projection (2DNPP), and then describes the problem of bidirectional two-dimensional approaches to feature extraction, and derives solution using the proposed B2DNPP. The architecture and design of the ELM classifier is presented in Section 4, and then the facial recognition methodology proposed for the C-B2DNPP framework is illustrated in detail in Section 5. A comparative study and experimental results are subsequently presented in Section 6.

## **4.1 Introduction**

The facial images subjected to a dimensionality reduction technique cannot provide an effective facial feature set [127]. Therefore, over the last few decades, approaches based on transforms have often been implemented as a pre-processing step in dimensionality reduction, in order to achieve a high level of discrimination among classes in pattern recognition and face recognition applications. The recent development of different multi-resolution tools has prompted research to deploy these techniques to improve performance in facial recognition applications. In these enhanced techniques, facial images are first transformed into a new space through one of the tools of multi-resolution analysis (such as the wavelet or curvelet), and then a dimensionality reduction algorithm such as in PCA, LDA, LPP and NPP is used in order to generate representative facial features and to classify the features optimally. The most widely used tool of multi-resolution analysis for feature extraction is the wavelet transform, which is popular in face recognition [128, 129]. The wavelet tool decomposes 1-D vectors or 2-D images at different resolutions and levels using high pass- and low pass-filters. In the wavelet transform, an image is decomposed into four sub-bands

corresponding to approximation (L1)-, horizontal (H1)-, vertical (V1)- and diagonal (D1)- sub-bands. The (L1) approximation component can be further decomposed into four further sub-bands and so on [130], as shown in Figure 4.1. A recent survey [131] contains details of the theory of wavelet transforms for face recognition applications.

The wavelet transform has been successfully applied to recent problems in different fields, such as image processing and computer vision. In recent years, researchers have implemented various wavelet-based face recognition frameworks. For example the images are first decomposed into different frequency sub-bands using the wavelet transform, and the mid-range frequency components are used later for principle component analysis (PCA) representation [130]. Other researchers [132] have performed linear discriminant analysis (LDA) on the approximation component wavelet-faces extracted by the wavelet transform from facial images. Alternatively, the facial images are inputted into two levels of the wavelet transform [133], and then approximation wavelet sub-bands are used in the HMM classifier. In another study [134], the face images were decomposed into two levels using the wavelet transform system, followed by the projection of the approximated wavelet coefficients using PCA using a support vector machine (SVM) classifier for recognition. Zange et al [135] proposed a modular face recognition scheme called the (*k*AM model), which decomposes the input facial images using the wavelet transform, after which it combines the lower resolution sub-band representations and uses kernel associative memory techniques. Wavelet analysis has been also employed to decompose each input image into 24 different sub-bands [136]. In the first level, the raw face image is decomposed into four wavelet components (L1, H1, V1, and D1). Then in level 2 only the L1 sub-band is further decomposed to obtain (L2, H2, V2, and D2) components. After that in level three, all of these sub-bands are decomposed to obtain 24 different sub-band images from each input facial image. These decomposed sub-band images are then projected into a smaller subspace using principal and independent component analysis (PCA and ICA). In similarly, a local discriminant coordinates (LDC) method has been implemented [137] based on a wavelet sub-bands package, and in this study a new dilation invariant entropy used with a maximum posteriori (MAP) logistic model was proposed. Here, the wavelet package is obtained in the same way as previously [136], but this time only using the last 16 decomposed wavelet components. A new algorithm was then introduced [138] named the wavelet transform weighted modular

PCA (WTWMPKA), which combines the wavelet sub-bands with PCA. This method uses only the L2 wavelet component and the average of the H1 and V1 components.

The typical ideal image feature extraction technique should be multi-resolution, directional and local. Although the wavelet transform has the capability to localize in both the time and scale domains, but mode of functioning places limits on directionality [139]. Human faces do not always display isotropic scaling, and also contain curves and line features in different directions, and the wavelet will be unable to capture these distinctive features because it focuses only on the details of images with primarily horizontal, vertical and diagonal orientation. Therefore, it has only a fixed number of directional elements that are independent of scale. and ignores other directional information which is important in human face recognition systems [54]. These limitations have motivated researchers to develop a new multi-resolution transform that meets the needs of image analysis.

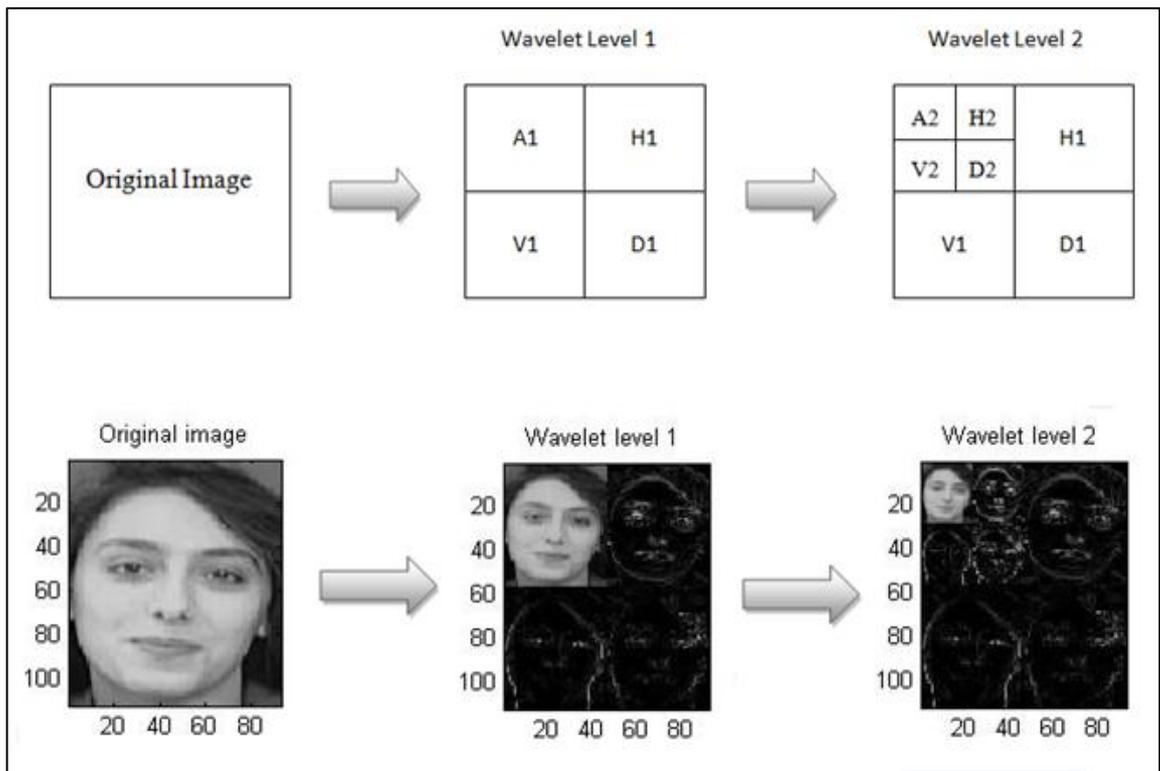


Figure 4.1: Two levels wavelet decomposition of a two-dimensional facial image

In 2000, Candes and Donoho [126, 140-142] developed a new tool for multi-resolution analysis. The well-known curvelet transform overcomes the limitations of the

wavelet transform mentioned above by enhancing the representation of edges and other singularities along curves and improving the capture of directional features [139]. It has inspired researchers to apply it successfully in different image processing field such as image fusion [143, 144], enhancement [145], compression [146, 147], denoising [140], high quality image restoration [148] and texture classification [149]. Curvelet transforms have also been used in the fields of pattern recognition and face recognition. Zhang et al. [150] extracted curvelet coefficients using the curvelet transform, and then employed a support vector machine (SVM) for classification. Tanaya et al [151] trained three SVMs using the curvelet coefficient and made decisions by simple majority voting. In this method, features were extracted from quantized images using the curvelet transform. Majumdar and Bhattacharya [152] performed the curvelet transform on 2-bit, 4-bit and 8-bit image representations at five different resolutions, then the 15 feature coefficients extracted (the 3 different image representations with 5 curvelet resolutions each) are used to train 15 SVMs. Other researchers [153] introduced a method that applies PCA to the curvelet approximated coefficients. Rziza et al proposed the combination of curvelet sub-bands with LDA [154]. They divided the curvelet coefficients into sub-blocks, and calculated various statistical measures such as entropy, variation and means for each sub-block. Finally, they constructed feature vectors by concatenating the values for each block and using these vectors as input to the LDA. Xie in [155] integrated the curvelet transform with the least square support vector machine (LS-SVM). He employed the curvelet transform to decompose face images, and trained the LS-SVM using curvelet features. Mohammed et al [139] implemented a bidirectional two-dimensional PCA (B2DPCA) method based on approximated curvelet coefficients [54].

All existing methods that have utilized the curvelet transform and feature extraction for face recognition application are sensitive to outliers [32], because they have the ability to reveal only the global and spatial structure of samples on the data manifold. In pattern recognition, however, local information is more important than global structure [32]. Therefore, a novel feature extraction method called C-B2DNPP is proposed which combines the curvelet transform with bidirectional two-dimensional neighborhood preserving projection (B2DNPP) for face recognition in aiming to preserve the local neighbourhood structure on data space. In order to improve the performance of the

system, the extreme learning machine (ELM) neural network method [165] is used. The next section explains in detail the theory of the curvelet transform.

## 4.2 Curvelet Multi-resolution Transform

This section briefly explains the theory of the curvelet transform as well as its implementation. Additional mathematical details about the curvelet transform are available elsewhere [156].

### 4.2.1 Theory of curvelet transform

The curvelet transform is one of the most interesting approaches to multi-scale image representation [142], and was designed to overcome the limitations of existing multi-resolution representations. Compared with the use of other multi-resolution techniques which result in loss of information, such as the Gabor transform [157], the curvelet transform was developed to fully capture information about frequency and discrete domains. This reveals information about directions and represents curves as functions of different lengths and widths following the law of scaling parabolic  $width \cong (length)^2$ . The curvelet analysis method called second dyadic decomposition (SDD) is shown in Figure 4.2 . The curvelet transform is defined in discrete domains (Figure 4.2 (a)) using concentric squares and shears, while concentric circles and rotations are used to define the curvelet in the continuous domain (Figure 4.2 (b)). Hence, the figure explains the division of the curvelet frequency plane in radial (concentric squares or circles) and angular (rotations or shears) divisions into wedges (gray area in Figure 4.2). Radial division results in the decomposition of images in different scales, where the smallest scales represent the finest resolution whilst the coarsest resolution is defined as the largest scale. The angular division is responsible for the partition of each scale into different orientations or angles. The smallest number of orientations can be found at the coarsest resolution, while the maximum number of orientations can be found at the finest resolution. Therefore, curvelet transform units (wedges) are defined easily by their scale and orientation, where the length of each wedge is doubled at every other dyadic scale [156].

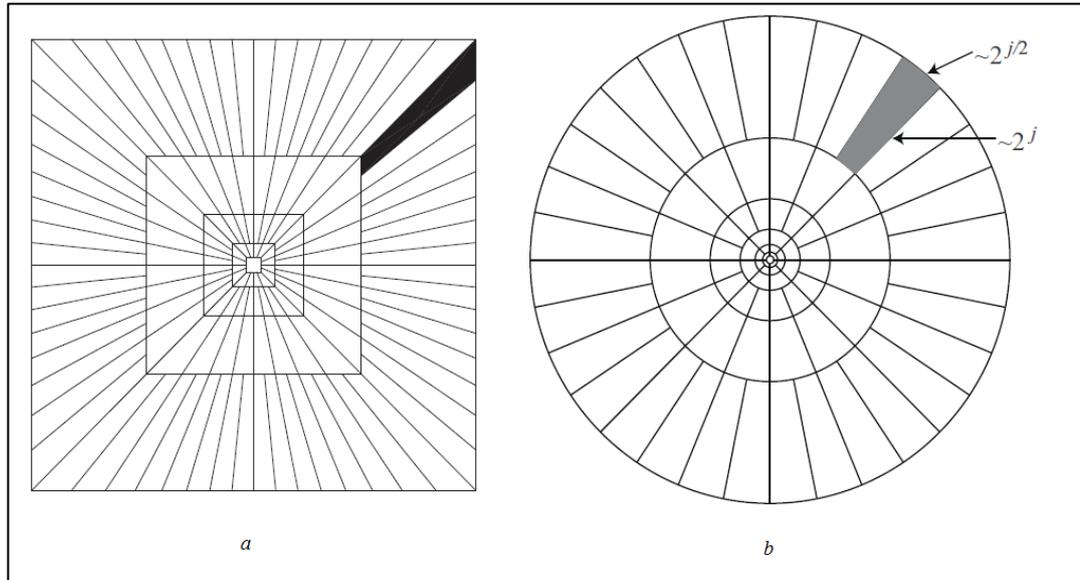


Figure 4.2: Curvelet space-frequency tiling, (a): discrete domain, (b): continuous domain. The gray shaded areas represent wedges.

In order to define the curvelet transform mathematically in the continuous domain, let  $x$  be a 2-D spatial variable in space  $R^2$ ,  $\omega$  is a frequency domain variable, and  $r$  and  $\theta$  are polar coordinates in frequency domain. Also  $\mathbf{W}(r)$  is a radial window used for radial partition which takes positive arguments and is supported on  $r \in [1/2, 2]$ , and  $\mathbf{V}(t)$  is an angular window used for angular partition that takes real arguments and is supported  $t \in [-1, 1]$ . Both windows  $\mathbf{W}$  and  $\mathbf{V}$  obey the following admissibility conditions to ensure a tight frame property:

$$\sum_{j=-\infty}^{\infty} \mathbf{W}^2(2^j r) = 1, r \in (3/4, 3/2), \quad (4.1)$$

$$\sum_{j=-\infty}^{\infty} \mathbf{V}^2(t - 1) = 1, t \in (-1/2, 1/2). \quad (4.2)$$

For each  $j \geq j_0$ , a frequency window  $\mathbf{U}_j$  is defined in the Fourier domain as:

$$\mathbf{U}_j(r, \theta) = 2^{-3/4} \mathbf{W}(2^{-j} r) \left( \frac{2[j/2]\theta}{2\pi} \right), \quad (4.3)$$

where,  $[j/2]$  is the integer part of  $\frac{j}{2}$ . Therefore, the support of  $\mathbf{U}_j$  is a polar wedge defined by the support of the windows  $\mathbf{W}$  and  $\mathbf{V}$  at a given scale.

A waveform  $\varphi_j(x)$  can be defined by its Fourier transform as  $\hat{\varphi}_j(\omega) = \mathbf{U}_j(\omega)$ . So, all curvelets at scale  $2^{-j}$  are obtained by the rotation and translation of  $\varphi_j$ . The curvelet transform is defined as a function of  $x = (x_1, x_2)$  at scale  $2^{-j}$ , orientation  $\theta_t$ , and position  $x_k^{(j,l)} = \mathbf{R}_{\theta_t}^{-1}(\mathbf{k}.2^{-j}, \mathbf{k}_2.2^{-j/2})$  by :

$$\varphi_{j,l,k}(x) = \varphi(\mathbf{R}_{\theta_t}(x - x_k^{(j,l)})), \quad (4.4)$$

where  $\theta_t = 2\pi.2^{[j/2]}.l$ ,  $l = 0, 1, \dots$ ,  $0 < \theta_t < 2\pi$ ,  $\mathbf{R}_{\theta_t}$  is a rotation matrix by  $\theta$ ,  $\mathbf{R}_{\theta_t}^{-1}$  is the inverse of  $\mathbf{R}_{\theta_t}$ , and  $k = (k_1, k_2)$  is the shift parameter. Thus, the continuous-time curvelet coefficient is determined by calculating the inner product of an element  $f \in L^2(\mathbb{R}^2)$  and curvelet  $\varphi_{j,k,l}$  :

$$c_{j,k,l} = \langle f, \varphi_{j,k,l} \rangle = \int_{\mathbb{R}^2} f(x) \overline{\varphi_{j,k,l}(x)} dx. \quad (4.5)$$

Plancherel's theory is applied to Equation 4.5, in order to express  $c_{j,k,l}$  as an integral over the entire frequency plane, so that  $c_{j,k,l}$  becomes:

$$c_{j,k,l} = \frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} \hat{f}(\omega) \overline{\hat{\varphi}_{j,k,l}(\omega)} d\omega = \frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} \hat{f}(\omega) \mathbf{U}_j(\mathbf{R}_{\theta_t}\omega) e^{i\langle x_k^{(j,l)}, \omega \rangle} d\omega. \quad (4.6)$$

The curvelet transform has additional coarse scale elements, which can be defined as:

$$\varphi_{j_0,k}(x) = \varphi_{j_0k}(x - 2^{-j_0})k, \hat{\varphi}_{j_0}(\omega) = 2^{-j_0} \mathbf{W}(2^{-j_0}|\omega|). \quad (4.7)$$

Equation 4.5 can be expressed to the discrete curvelet that can be the form used for an image  $f[x_1, x_2](0 \leq x_1, x_2 < n)$  in Cartesian space as follows:

$$c_{j,k,l}^D = \sum_{0 < x_1, x_2 < n} f[x_1, x_2] \overline{\varphi_{j,k,l}^D[x_1, x_2]}. \quad (4.8)$$

### 4.2.2 Implementation of the curvelet transform

In general, there are two generations of the curvelet transform. The first generation was proposed by Emmanuel Candès and David Donoho in 1999 [126]. The implementation of this generation is defined the curvelet transform as a new transform between wavelet and ridgelet transform. It can be summarized in four main steps as follows:

1. An object  $f[t_1, t_2], t_1 \geq 0, t_2 \leq n$  is decomposed first into sub-bands by the wavelet transform in order to obtain  $\hat{f}[t_1, t_2]$ .
2. Each sub-band is windowed into squares of a suitable scale of side-length  $2^{-j}$ , as shown in Figure 4.2.
3. Each square obtained is then renormalized to unit length and width  $2^{-j}$ .
4. Finally, each square is decomposed using the ridgelet transform.

The second generation of the curvelet transform was proposed in 2002, and then Candès and Donoho developed its implementation in 2006. The new transform is faster, simpler, and involves less redundancy than the first generation [156]. Two different digital implementations of the fast digital curvelet transform (FDCT) were defined as follows:

1. The curvelet transform based on the unequally spaced fast Fourier transform (USFFT).
2. The curvelet transform based on the wrapping of specially selected Fourier samples.

Both of these implementations are linear and take the Cartesian matrix as input, but they differ in the choice of spatial grid used to translate the curvelets at each scale and angle. A tilted grid aligned with the axes of the window leads to the USFF, while a grid

aligned with the input Cartesian grid leads to wrapping. Both transformation methods provide output consisting of discrete coefficients, but the wrapping method is a more intuitively appropriate approach and requires less computational time [156]. Therefore, this chapter adopt the FDCT wrapping method to extract curvelet coefficients. This method can be described as follows:

Let  $f[t_1, t_2], t_1 \geq 0, t_2 \leq n$  be an image object in Cartesian space, and  $\hat{f}[n_1, n_2], -n/2 \geq n_1, n_2 \leq n/2$  represents its 2-D discrete Fourier transform. Let  $\mathbf{U}(\omega)$  denote a discrete localizing window and  $\tilde{\mathbf{U}}_j[n_1, n_2]$  is supported on some rectangle of length  $L_{1j}$  and width  $L_{2j}$  [156]. Then there are four steps in the implementation of curvelets via wrapping, as follows:

1. Apply the 2-D fast Fourier transform (FFT) to the image object  $f[t_1, t_2]$  and obtain Fourier samples  $\hat{f}[n_1, n_2]$ .
2. Form the product  $\tilde{\mathbf{U}}_{j,l}[n_1, n_2] \hat{f}[n_1, n_2]$  for each scale  $j$  and angle  $l$ , where  $\tilde{\mathbf{U}}_j[n_1, n_2]$  is the discrete localizing window ( as shown in Figure 4.3 (a)).
3. Wrap this product around the origin in order to obtain  $\hat{f}_{j,l}[n_1, n_2] = \mathbf{W}(\tilde{\mathbf{U}}_{j,l} \hat{f})[n_1, n_2]$  where the ranges of  $n_1, n_2$  and  $\theta$  are  $0 \leq n_1 < L_{1j}, 0 \leq n_2 < L_{2j}$ , and  $-\pi/4, \pi/4$  respectively;  $L_{1j} \approx 2^j$  and  $L_{2j} \approx 2^{j/2}$  are constants (Figure 4.3 (b)).
4. Apply the inverse 2-D FFT to each  $\hat{f}_{j,l}$ , hence obtaining the discrete curvelet coefficients  $c_{j,k,l}^D$ .

In the first two steps, an image is divided into radial and angular wedges that follow the relationship between a curvelet's length and width via the application of 2-D FFT to an image. Then, the data is re-indexed around the original during step three. Lastly, in step four, the inverse 2-D FFT is applied in order to create the discrete curvelet coefficients in the spatial domain. In the proposed method, all of the curvelet coefficients in each curvelet sub-band are used independently as initial features to represent images in order to reveal local information in the image objects.

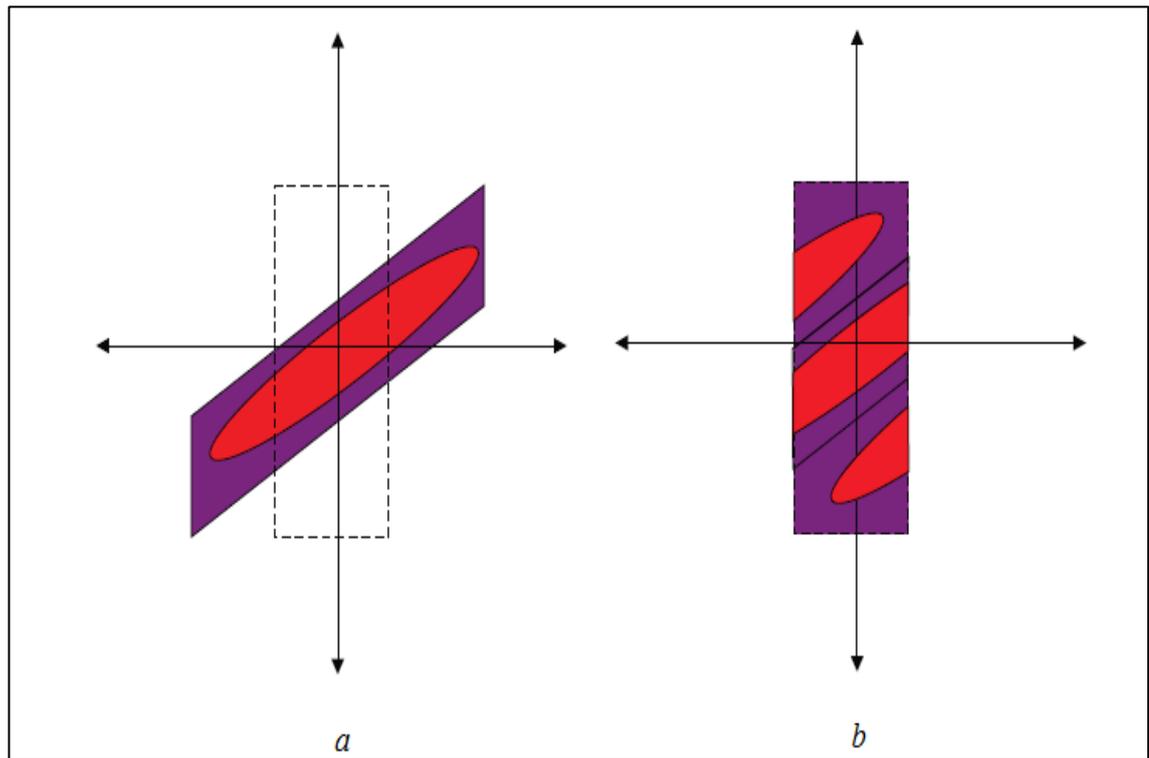


Figure 4.3: Wrapping a segment around a discrete localization window: (a) the original before wrapping and (b) after wrapping.

### 4.3 Bidirectional Two-Dimensional Neighborhood Preserving Projection Based on Curvelet Transform (C-B2DNPP)

This section presents a novel manifold learning method, called bidirectional two-dimensional neighborhood preserving projection based on the curvelet transform for face recognition. In this method, the curvelet transform is applied first to reveal the features of a facial image at different scales and angles. Then, a new B2DNPP is applied to curvelet coefficients obtained. This is performed independently for each curvelet sub-band in order to preserve the local structure of the data samples. For better recognition accuracy, all of the features extracted by B2DNPP for each sample in one vector are concatenated and fed into the ELM classifier in the classification step. Because the B2DNPP is an extension of the 2DNPP method, the latter is briefly reviewed here. Then, the problem of bidirectional two-dimensional method is described followed by detailed derivation of the B2DNPP. For ease of discussion in the following sections, the 2DNPP [50] is termed the column-direction 2DNPP.

### 4.3.1 The column-direction 2DNPP

In 2012, Zhang et al. [50] proposed a two dimensional neighbourhood preserving projection method (2DNPP) for face recognition, which extracts features directly using 2-D image matrices instead of converting there into long 1-D vectors. 2DNPP is derived from the NPP method [33], and uses the same NPP procedure to build the weight matrix from the nearest neighbor affinity graph. This method improves the accuracy rate and reduces computational time compared with the 1-D NPP method. Moreover, it overcomes the limitation of under-sample size problem when the number of images sampled is lower than that of their dimensions. The mathematical implementation of the 2DNPP algorithm is given below:

Let  $\{\mathbf{X}\} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_M] \in \mathbb{R}^{I_1 \times I_2}$  be a dataset of  $M$  training images. Each image  $\mathbf{X}_m$  is projected into  $\mathbf{Y}_m \in \mathbb{R}^{P_1 \times I_2}$  in a lower-dimensional space which using the transformation function  $\mathbf{Y}_m = \mathbf{U}_1^T \mathbf{X}_m$  where  $\mathbf{U}_1 (\mathbf{U}_1 \in \mathbb{R}^{I_1 \times P_1}, P_1 < I_1)$ , is a linear transformation matrix. And let  $\mathbf{W} = [w_{ij}] \mathbb{R}^{M \times M}$  represent a weight matrix used to reconstruct each image from its  $k$  nearest neighbors. The projected image  $\mathbf{Y}_i$  can be obtained by minimizing the following function:

$$\Phi(\mathbf{Y}) = \left\| \sum_i \mathbf{Y}_i - \sum_j w_{ij} \mathbf{Y}_j \right\|_2^2. \quad (4.9)$$

Equation 4.9 aims to minimize the value of  $\Phi(\mathbf{Y})$  associated with  $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_M] \in \mathbb{R}^{P_1 \times (I_2 M)}$ . The objective  $\Phi(\mathbf{Y})$  can be rewritten as follows:

$$\begin{aligned} \Phi(\mathbf{Y}) &= \left\| \sum_i \mathbf{Y}_i - \sum_j w_{ij} \mathbf{Y}_j \right\|_2^2 = \left\| \mathbf{Y} - \mathbf{Y}(\mathbf{W} \otimes \mathbf{I}_{I_2}) \right\|_2^2 = \left\| \mathbf{Y}(\mathbf{I} \otimes \mathbf{I}_{I_2} - \mathbf{W}^T \otimes \mathbf{I}_{I_2}) \right\|_2^2 \\ &= \text{tr}\{\mathbf{Y}(\mathbf{I} - \mathbf{W})^T \otimes \mathbf{I}_{I_2} [(\mathbf{I} - \mathbf{W}) \otimes \mathbf{I}_{I_2}] \mathbf{Y}^T\} = \text{tr}\{\mathbf{Y}(\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W}) \otimes \mathbf{I}_{I_2} \mathbf{Y}^T\} \\ &= \text{tr}\{\mathbf{U}_1^T \mathbf{X}(\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W}) \otimes \mathbf{I}_{I_2} \mathbf{X}^T \mathbf{U}_1\} \end{aligned} \quad (4.10)$$

where  $\otimes$  is the Kronecker product,  $\mathbf{I}$  is the identity matrix of order  $M$ , and  $\mathbf{I}_{I_2}$  is the identity matrix of order  $I_2$ . In order to produce an orthogonal projection set  $\mathbf{Y}$ , the

constraint  $\mathbf{U}_1^T \mathbf{X} \mathbf{X}^T \mathbf{U}_1 = \mathbf{I}_{P_1}$  is impose the objective function  $\Phi(\mathbf{Y})$ , where  $\mathbf{I}_{P_1}$  is the identity matrix of order  $P_1$ . Thus, the minimization problem is reduced to:

$$\arg \min_{\mathbf{U}_1^T \mathbf{X} \mathbf{X}^T \mathbf{U}_1 = \mathbf{I}_{P_1}} \text{tr}\{\mathbf{U}_1^T \mathbf{X}[(\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W}) \otimes \mathbf{I}_{L_2}] \mathbf{X}^T \mathbf{U}_1\} \quad (4.11)$$

The solution  $\mathbf{U}_1$  to the optimization problem is given by the minimum eigenvectors of the generalized eigenvalue problem as follows:

$$\mathbf{X}[(\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W}) \otimes \mathbf{I}_{L_2}] \mathbf{X}^T \mathbf{U}_1 = \lambda \mathbf{X} \mathbf{X}^T \mathbf{U}_1 \quad (4.12)$$

where  $\lambda$  represents the eigenvalue solution of the problem, and  $\mathbf{U}_1$  is the basis of the eigenvectors corresponding to the  $P_1$  smallest generalized eigenvalues. Thus it is clear that the column-direction 2DNPP only operates along the column direction of the images and ignores information in the row-direction.

### **4.3.2 Bidirectional two-dimensional neighborhood preserving projection (B2DNPP)**

Actually, 2DNPP preserves the relationships between image columns and performs compression only in one direction. However, it does preserve variations between image columns, variations between image rows are also useful in pattern recognition applications [158, 159]. In contrast to 2DNPP, B2DNPP can effectively reveal the local information about variations in columns as well as rows for 2-D images, so it takes full advantage of structural information in the original space within facial images. Moreover, the proposed B2DNPP approach performs reduction in both the row- and column- directions of the image, and therefore it extracts the optimal number of features of images in lower-dimensional space and with fewer coefficients, as well as improving the preservation of neighborhood relationships. Consequently, B2DNPP provides much better projection matrices from 2-D facial images than either the NPP or 2DNPP methods.

#### 4.3.2.1 Formulation of the problem of dimensionality reduction

Consider  $\{\mathbf{X}\}$  as a facial dataset consisting of  $M$  training images ( $\mathbf{X}_m \in \mathbf{R}^{I_1 \times I_2}, m = 1, 2, \dots, M$ ). The goal of the bidirectional two-dimensional reduction method is to find,  $\mathbf{U}_1 \in \mathbf{R}^{I_1 \times P_1}, (P_1 < I_1)$  and  $\mathbf{U}_2 \in \mathbf{R}^{I_2 \times P_2}, (P_2 < I_2)$ , which represent left- and right- projection matrices respectively with orthonormal columns. Each image  $\mathbf{X}_m$  is projected into a lower dimensional feature matrix  $\mathbf{Y}_m \in \mathbf{R}^{P_1 \times P_2}$  using the following transformation function:

$$\mathbf{Y}_m = \mathbf{U}_1^T \mathbf{X}_m \mathbf{U}_2, \quad (m = 1, 2, \dots, M) \quad (4.13)$$

where  $\mathbf{Y}_m$  represent the linear combination of its  $k$  nearest neighbors in the original dataset. Using Equation 4.13, each image  $\mathbf{X}_m$  in the original facial dataset can be replaced by  $\mathbf{Y}_m$ . Since the left- and right- projection matrices eliminate the noise in the original facial image space and preserve distinctive properties, therefore the projected features  $\{\mathbf{Y}\}$  give the optimal representation of the original facial dataset  $\{\mathbf{X}\}$ .

#### 4.3.2.2 The B2DNPP projection learning

The B2DNPP algorithm is based on a simple intuition, which starts by constructing an optimal weight matrix  $\mathbf{W}$  associated with image samples and their  $k$  nearest neighbors in the same way as NPP explained previously in Section 2.4.4. Then, the optimization problem is solved to obtain projection matrices along the column and row directions. Finally, the original training facial images  $\{\mathbf{X}\}$  are projected to obtain a new feature images set  $\{\mathbf{Y}\}$ , which is represented as a linear combination of images  $\{\mathbf{X}\}$  along their  $k$  nearest neighbors. As shown in Figure 4.4, the proposed B2DNPP dimensionality reduction method operates along 2-D image columns and image rows in order to capture useful structural information which can improve recognition accuracy.

B2DNPP creates the affinity weight matrix  $\mathbf{W}$  from the original facial dataset  $\{\mathbf{X}\}$  using the same procedure as in NPP. Then, the column-direction 2DNPP solves the optimization problem to derive the optimal projection matrix  $\mathbf{U}_1 \in \mathbf{R}^{I_1 \times P_1}$  which reveals information about image columns. In the same way, the row-direction 2DNPP then

derives the optimal projection matrix  $\mathbf{U}_2 \in \mathbb{R}^{I_2 \times P_2}$ , preserving information about image rows. When the two projection matrices  $\mathbf{U}_1$  and  $\mathbf{U}_2$  have been obtained, every  $(I_1 \times I_2)$  image in the training dataset  $\{\mathbf{X}\}$  is projected onto  $\mathbf{U}_1$  and  $\mathbf{U}_2$  simultaneously to produce the feature set  $\{\mathbf{Y}\}$  using the transformation function in Equation 4.13. Hence, the new dataset  $\{\mathbf{Y}\}$  contains  $M$  image features of size  $(P_1 \times P_2)$  which are represented by fewer coefficients.

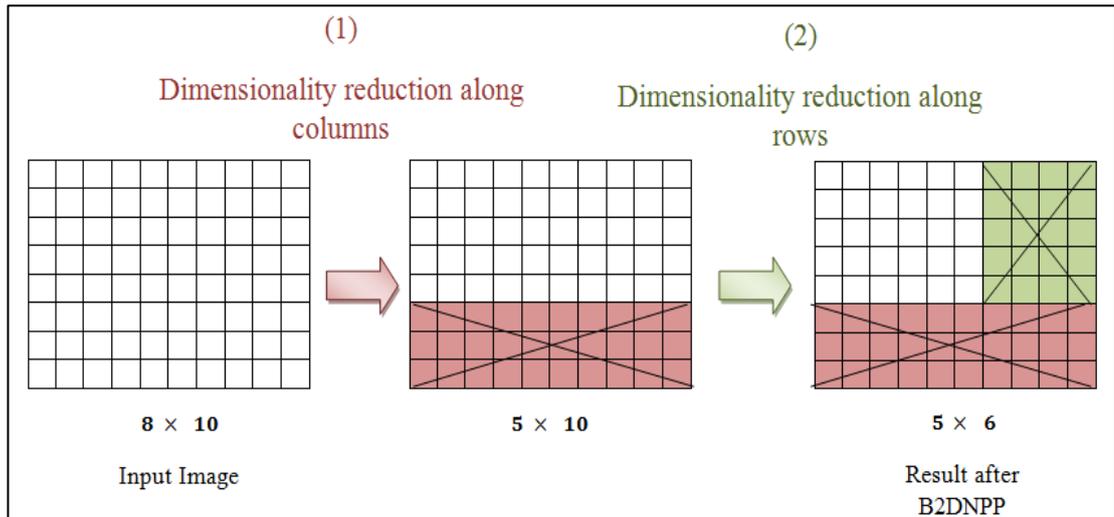


Figure 4.4: Block diagram of the proposed B2DNPP method

Put more simply the main implementation steps for the B2DNPP algorithm are as follows:

Input: input facial training images set  $\{\mathbf{X}\} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_M] \in \mathbb{R}^{I_1 \times I_2}$ , the number of nearest neighbor points ( $k$ ), and the new row- and column-dimensions for the reduced space ( $P_1$  and  $P_2$ ).

Output: feature image set  $\{\mathbf{Y}\}$ ,  $\mathbf{Y}_i \in \mathbb{R}^{P_1 \times P_2}$ .

Algorithm Steps:

1. Construct nearest neighbors graph by determining the  $k$  nearest neighbors of each facial image  $\mathbf{X}_i$  using the  $k$ -nearest neighbors method.
2. Build the affinity weight matrix  $\mathbf{W}$  by calculating the weights  $w_{ij}$  corresponding to each image in the dataset  $\{\mathbf{X}\}$  and its  $k$  nearest neighbors (Equation 2.19).

3. Obtain left- and right-projection matrices  $\mathbf{U}_1$  and  $\mathbf{U}_2$ , by solving the generalized eigenvectors problem for the column-direction 2DNPP and the row-direction 2DNPP as follows:

Column-direction 2DNPP:

The left optimal matrix  $\mathbf{U}_1$  can be found through solve Equation 4.12. Construct the projection matrix  $\mathbf{U}_1 \in \mathbf{R}^{I_1 \times P_1}$  to be consisting of eigenvectors corresponding to the  $P_1$  smallest eigenvalues.

Row-direction 2DNPP:

As with the derivation of the column-direction 2DNPP shown in Section 4.3.1 above, the minimization problem of the row-direction 2DNPP can be summarized as:

$$\arg \min_{\mathbf{U}_2^T \mathbf{X} \mathbf{X}^T \mathbf{U}_2 = \mathbf{I}_{P_2}} \text{tr}\{\mathbf{U}_2^T \mathbf{X}[(\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W}) \otimes \mathbf{I}_{I_1}] \mathbf{X}^T \mathbf{U}_2\} \quad (4.14)$$

where,  $\mathbf{I}_{I_1}$  is the identity matrix of order  $I_1$ , and  $\{\mathbf{X}\} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_M] \in \mathbf{R}^{I_1 \times I_2}$ . Then, an optimal matrix  $\mathbf{U}_2 \in \mathbf{R}^{I_2 \times P_2}$  that contains  $P_2$  eigenvectors associated with the  $P_2$  smallest eigenvalues, can be obtained by solving the following generalized eigenvectors problem:

$$\mathbf{X}[(\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W}) \otimes \mathbf{I}_{I_1}] \mathbf{X}^T \mathbf{U}_2 = \lambda \mathbf{X} \mathbf{X}^T \mathbf{U}_2. \quad (4.15)$$

4. Calculate the feature matrices  $(\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_M)$  of all training facial images  $(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_M)$  in low dimensional space using the linear transformation function  $\mathbf{Y}_m = \mathbf{U}_1^T \mathbf{X}_m \mathbf{U}_2$ , ( $m = 1, 2, \dots, M$ ).

The above algorithm describes the unsupervised form of the B2DNPP algorithm, which does not need any information related to facial image classes. As in the NPP and 2DNPP methods, it is easy to extend B2DNPP to a supervised version when labels for the training face images are available. This is achieved by selecting the value of  $k$ , which is the number of nearest neighbors corresponding to the number in the training

sample  $\mathbf{X}_i$ . Hence, the nearest neighbor images of a data sample  $\mathbf{X}_i$  are only considered if they belong to the same class as  $\mathbf{X}_i$ . Therefore, supervised B2DNPP can combine discriminative information in the weighting of the images.

#### **4.4 Extreme Learning Machine (ELM) Classifier**

Among pattern recognition techniques, artificial neural networks ANNs have been increasingly used as an alternative way to implement basic pattern classifiers such as KNN classifiers [160]. ANNs can be viewed as systems inspired by the operation of biological neural networks. Their learning systems consist of enormous numbers of highly interconnected artificial neurons. These neurons process information according to their self-learning capability. The major characteristic of ANNs is that they have the capability to achieve a non-linear approximation function, which describes the interrelationship between dependent and independent features present in the input data samples [161]. Therefore, ANNs are extensively used in pattern recognition, classification and regression problems. There are three main advantages of using ANN systems: (1) they can perform classification tasks that a linear classifier cannot; (2) the network can continue based on its parallel nature even if one element fails to operate; and (3) the ANN's learning does not need to be reprogrammed. However, ANNs is suffer from some limitations such as needing to be trained before use, problems with the selection of the number of hidden neurons, over-fitting, and the long processing times required for a large network [162].

Over the past two decades, many neural network learning systems have been developed, such as the back-propagation (BP) algorithm which is designed to be used with single-layer and multi-layer models. This is widely used in different applications such as image processing, pattern recognition, feature selection and others. Although, it is one of the most popular ANN systems, it involves high communication costs, and is computationally slow [163]. Recently, Huang et al [164] proposed a new learning system called the extreme learning machine (ELM) intelligence technique, which can be used to train a single hidden layer feedforward neural network (SLFN) architecture as shown in **Error! Reference source not found.** The ELM has been used widely in various applications such as biometrics, bioinformatics, image processing, signal processing, human action recognition, security and data privacy, and human computer interfacing. The ELM method outperforms the BP method as well as overcoming some

of the limitations of ANNs and support vector machines SVMs, such as poor computational ability and slow learning speed [165].

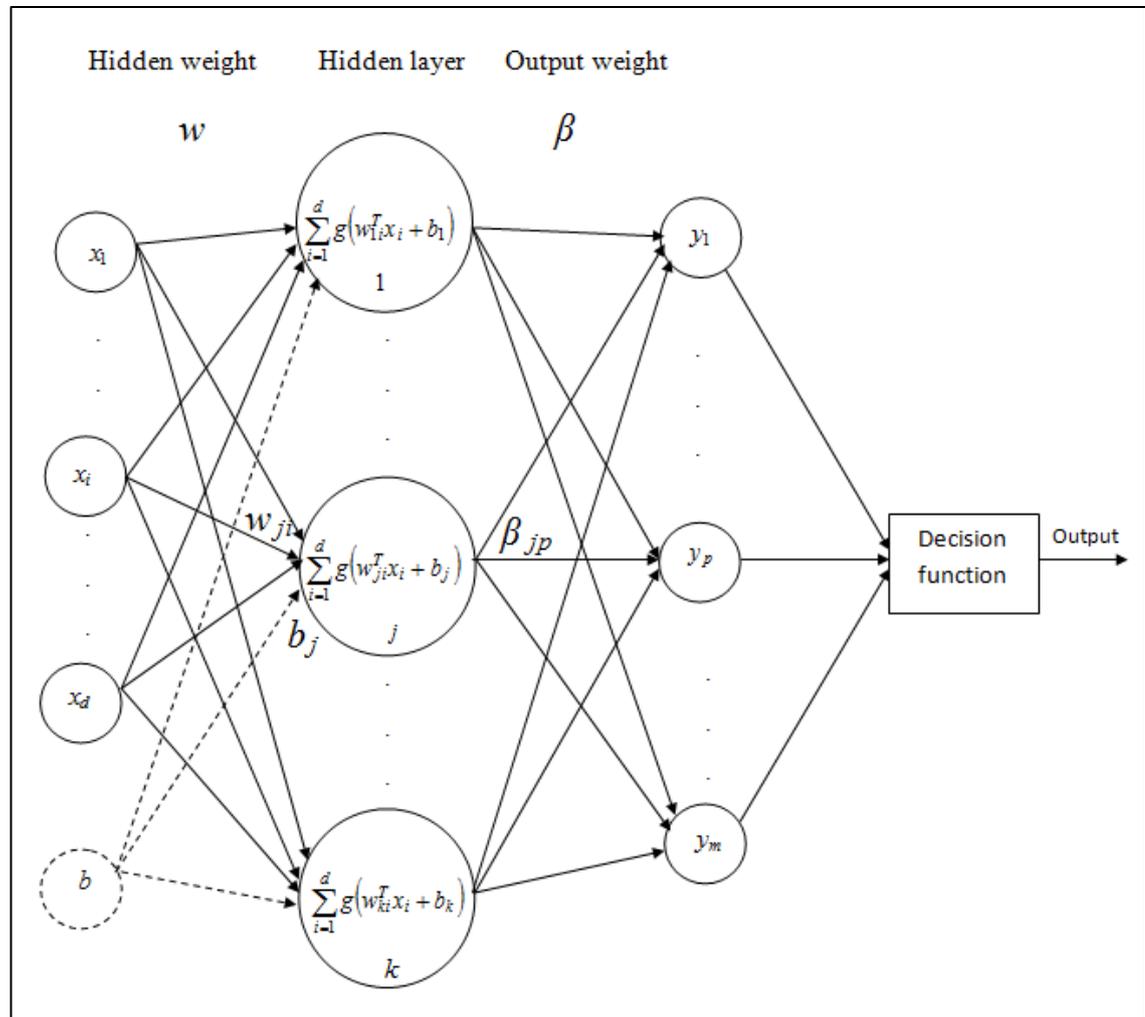


Figure 4.5 : The architecture of an ELM classifier

The ELM technique has been attracting attention due to its fast and simple training algorithm, which randomly selects input weights. In the ELM algorithm, the input weights and the biases of the hidden layer are randomly selected without training, which transforms the training of the SLFN into a linear equation system in terms of connecting the hidden-layer to the output-layer through the use of output weights. The solution for this linear system is obtained using the Moore-Penrose generalized pseudo inverse [166]. Therefore, the output weights can be determined by a simple generalized inverse operation of the hidden layer output. Consequently, the learning speed of the ELM can be several times faster than the BP which iteratively tunes the parameters [167].

The Least-square based approach [164] (e.g. radial basis function (RBF) network learning) is usually used in the training of one hidden layer feedforward neural network with activation function  $g(x) : R \rightarrow R$ , (e.g. Gaussian)  $g(x) = \exp(-x^2)$ ,  $G(\mathbf{a}_i, b_i, \mathbf{x})$  is given by:

$$G(\mu_i, b_i, \mathbf{x}) = g(b_i \|\mathbf{x} - \mu_i\|), b_i \in R^+ \quad (4.16)$$

Where  $\mu_i$  and  $b_i$  are the center and impact factor of the  $i$ th RBF hidden node, and  $R^+$  refers to the set of all positive real values. The RBF network is a special case of SLFNs with RBF nodes in its hidden layer. Each RBF node has its own centroid and impact factor, and its output is given by a radially symmetric function of the distance between the input and the center.

The ELM algorithm setup and its steps can be briefly stated as follows [164]:

Input: Training dataset  $L = \{(x_i, t_i), i = 1, 2, \dots, N\}$ , where  $x_i \in R^d$  and  $t_i \in R^m$ ; the hidden neurons number  $K$ ; the activation function is  $g(\cdot)$ , here  $g$  is a radially symmetric kernel function. Such kernels include the popularly used Gaussian function:

$$g(\mu_i, b_i, x) = \exp\left(-\frac{\|x - \mu_i\|^2}{b_i^2}\right),$$

where  $\mu$  is the  $i$ th kernel's center and  $b$  is its impact width.

Output: The single hidden layer feedforward neural network (SLFN) with the determined output weight vectors  $\beta_i \in R^m$ , where  $\beta_i$  is the weight vector that connects the  $i$ th hidden neuron to the output layer.

Algorithm Steps:

1. Assign the weight vector  $w_i \in R^d$  and bias  $b_i \in R$  randomly, where  $w_i$  is the weight vector that connects the input layer to the  $i$ th hidden neuron, and  $b_i$  is the bias of the  $i$ th hidden neuron for  $i = 1, 2, \dots, k$ .
2. Determine the **H** matrix as follows:

$$\mathbf{H} = \begin{pmatrix} g(w_1^T x_1 + b_1) & \dots & g(w_K^T x_1 + b_K) \\ \cdot \\ \cdot \\ g(w_1^T x_N + b_1) & \dots & g(w_K^T x_N + b_K) \end{pmatrix} \in \mathbf{R}^{N \times K}. \quad (4.17)$$

3. Compute the Moore–Penrose generalized inverse of the matrix  $\mathbf{H}^+$  [166, 168] of the matrix  $\mathbf{H}$ .
4. Compute the output weight vector  $\beta$  using the equation:

$$\beta = \mathbf{H}^+ \mathbf{T}, \quad (4.18)$$

where:

$$\beta = \begin{bmatrix} \beta_1^T \\ \cdot \\ \cdot \\ \beta_K^T \end{bmatrix} \in \mathbf{R}^{K \times m} \text{ and } \mathbf{T} = \begin{bmatrix} t_1^T \\ \cdot \\ \cdot \\ t_N^T \end{bmatrix} \in \mathbf{R}^{N \times m}. \quad (4.19)$$

The measured output value for any data sample  $x \in \mathbf{R}^d$  can be obtained by:

$$\tilde{y} = \sum_{i=1}^K \beta_i g(w_i^T x + b_i), \quad (4.20)$$

where  $\beta \in \mathbf{R}^m$  is the output of the ELM algorithm.

## 4.5 The Proposed C-B2DNPP Method for Face Recognition

The proposed C-B2DNPP method is divided into two phases, as follows:

### 4.5.1 C-B2DNPP training phase

The proposed C-B2DNPP algorithm involves a new technique for face feature extraction based on curvelet multi-resolution analysis. The resulting features are combined and inputted to the classifiers. This approach consists of three major stages as shown in Figure 4.6 and summarized as follows.

1. Pre-processing

During this stage, all of the training facial images are aligning, rotation and normalized to improve the recognition rate as explained previously in Section 3.7.

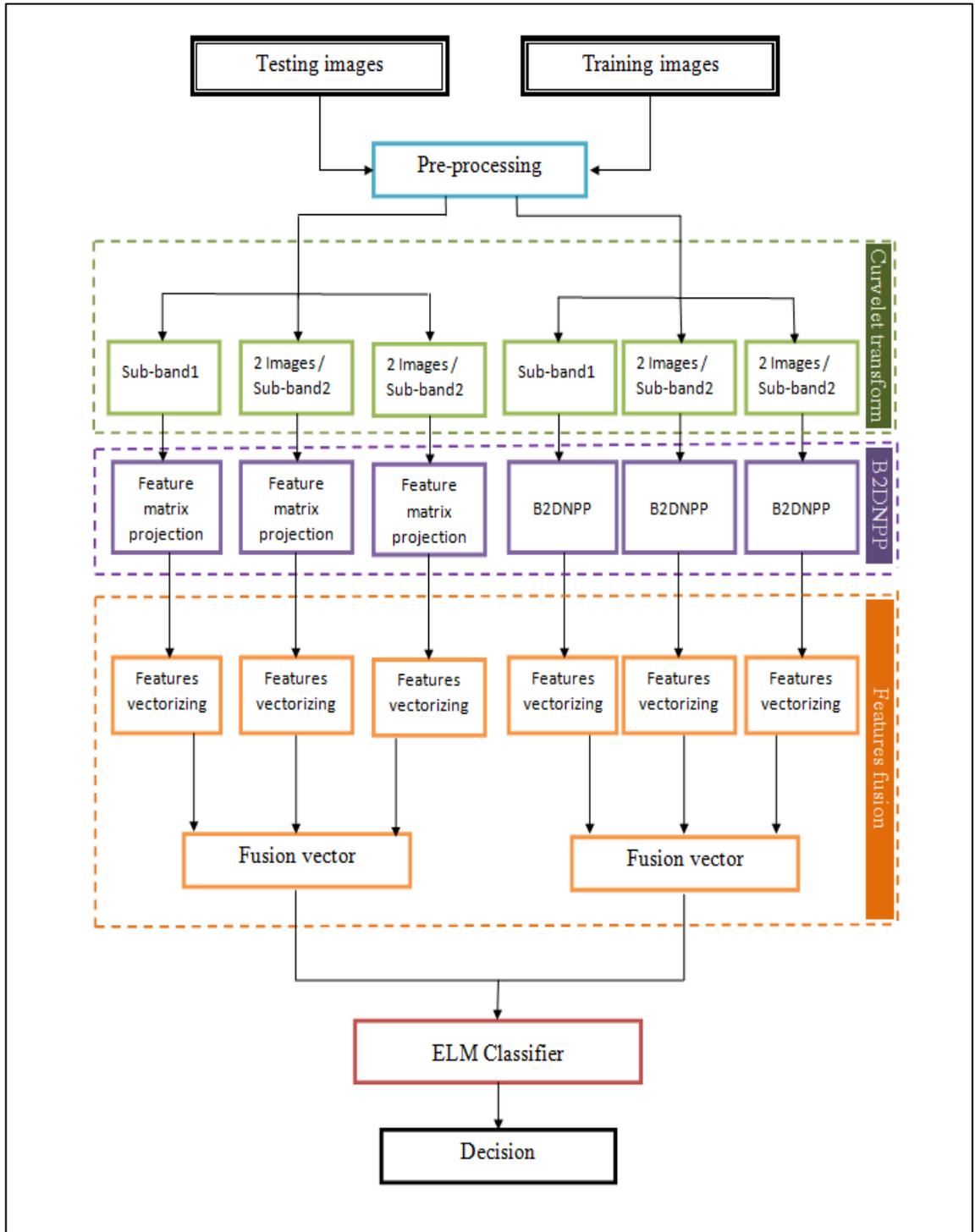


Figure 4.6: Methodology of the proposed C-B2DNPP method.

## 2. Feature extraction

The most critical step in successful facial recognition is the extraction of a useful feature set that can be used to distinguish between different persons. The appropriate extraction of a representative feature set can vastly improve the performance of the recognition algorithm. Therefore, the proposed feature extraction phase was implemented using two independent methods; the first used the curvelet multi-resolution transform and the second the proposed bidirectional two-dimensional neighborhood preserving projection (B2DNPP) method.

### a. Curvelet Transform

As discussed in the introduction to this chapter, the multi-scale curvelet transform has been proven to be helpful in pattern recognition. It has an effective capability representation to curved edge and good directional ability. This make it a robust tool for extracting edge information from facial images, especially that concerning smoothness along curves [139]. Therefore, in order to extract powerful features, the facial images are first decomposed into sub-bands at various angles and different scales using the curvelet transform. These sub-bands reveal a representative and efficient directional feature set. Figure 4.7 shows the curvelet coefficients of a facial image decomposed at scale = 4 (coarse, fine, finest), and two angles = 8, 16.

It can be noticed that the higher curvelet scales, such as six or even five, did not improve the recognition accuracy of the proposed method. Therefore, only four different scales have been used throughout this research in order to maintain a balance between the performance and speed of this system. When using the curvelet transform at scale= 4 and angle= 8, 26 decomposed coefficients are produced: 1 coarse (approximate) sub-band, 8 fine coefficients, 16 further finer coefficients, and one finest sub-band. In this work, the selection of curvelet sub-bands depends on the basis of the level of variation that they show. So, the sub-bands that demonstrate maximum variation are selected as initial image features to represent each original image, as shown in Tables 4.1-4.4 Thus, only the coarse sub-band (which contains the maximum energy of the data) in addition to the fine sub-band have been chosen. The finer sub-band and finest coefficient with the lowest amount of variation are ignored.

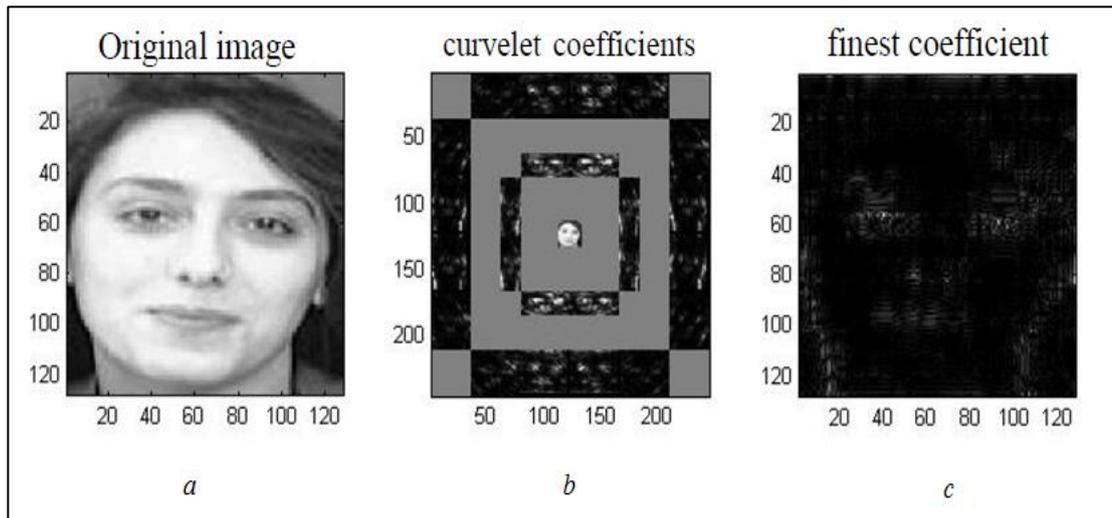


Figure 4.7: Example of curvelet coefficients at 4 scales: (a) original image, (b) coarse and fine curvelet coefficients (sub-bands 1-3), and (c) finest coefficient (sub-band4).

b. B2DNPP:

The B2DNPP method has the ability to preserve the manifold subspace of face images. As well as this, it performs a kind of dimensionality reduction. Therefore, it is applied to the selected curvelet sub-bands independently in order to produce an efficient and representative feature set.

3. Features fusion

After performing dimensionality reduction, each curvelet sub-band has its own B2DNPP features. The features in each sub-band are converted into a vector. Then, all of the feature vectors for the different sub-bands are combined into one vector. This fusion feature vector can be fed directly to the ELM classifier.

**4.5.2 C-B2DNPP recognition phase**

As shown in Figure 4.6, the C-B2DNPP recognition method that started with the testing images consists of 4 stages which are summarized as follows:

1. Pre-processing

The face area needs to be extracted, and normalized so as to be of standard size. Here, the testing images should follow the same pre-processing steps used for the training images.

2. Feature extraction

a. Curvelet transform

The testing images are decomposed using the curvelet transform and curvelet coefficients are obtained at two scales and one angle with the value of (8). Only the approximate coefficient and fine coefficients are selected to be used in the subsequent steps.

b. Projection

After the left- and right- projection matrices has been obtained in the training stage by the B2DNPP, each selected coefficient can be directly projected onto the lower dimension space using these two matrices in order to extract features from these sub-bands.

3. Feature fusion

Repeating the same fusion process as in the training phase, the feature matrices for the different sub-bands are vectorized first, and then combined together in one vector, which will be used in the classification step.

4. Classification

The classification step is performed by the ELM classifier which has become one of the most popular classifiers, as mentioned in the previous section. The training and test fusion feature vectors generated by the B2DNPP are used as input into the ELM classifier. Only one hidden layer is used. The number of nodes in the hidden layer should be less than or equal to the number of training samples [164], while the number of neurons in the output layers is automatically determined according to the number of the classes in the training vectors. The root mean square error (RMSE) is used to evaluate the prediction performance of the ELM classifier using the following formula:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \tilde{y}_i)^2}, \quad (4.21)$$

where  $N$  is the number of facial images in the testing set,  $y_i$  represents the actual class value of the testing image  $x$ , and  $\tilde{y}_i$  represents the expected class value of the same testing image.

## 4.6 Experiment Results

This section compares the facial recognition performance of the proposed two algorithms B2DNPP and C-B2DNPP, in addition to other popular two-dimensional and bidirectional two-dimensional techniques including 2DPCA [47], 2DLDA [48], 2DLPP [49], 2DNPP [50], (2D)2PCA [51], (2D)2FLD [52], B2DLPP [53], and curvelet+B2DPCA [54], are compared for facial recognition. After that, the effect of dimensionality reduction for the B2DNPP method is investigated. Three benchmark facial databases, ORL [117], AR [118] and FERET [119, 120], were used to evaluate the effectiveness of the proposed methods B2DNPP and C-B2DNPP, which was measured in terms of error rate using the following equation.

$$\text{Error rate} = \left( (t_{no} - c_{no}) \times 100 / t_{no} \right), \quad (4.22)$$

where  $t_{no}$  is the total number of testing images and  $c_{no}$  is the total number of correctly classified images.

### 4.6.1 Proposed algorithm and setting

The proposed method is based on the decomposition of images using the curvelet transform. Then, B2DNPP is applied to the curvelet coefficients to obtain distinctive feature vectors. These vectors are used later to classify the images using the ELM classifier to perform the identification task. The block diagram of the proposed method is shown Figure 4.6.

The original ORL images are greyscale, while images from both AR and FERET are colour. Therefore, in the present study, all facial images in AR and FERET datasets were converted into gray level images, represented by eight bits (256 gray levels). A cropping process was applied to all images in order to isolate the regions of interest. The cropped images were resized and normalized to  $80 \times 80$  pixel. The three datasets were randomly divided into training and testing subsets, such that 40% of the images for

each subject were used as gallery subset and 60% used as the probe subset. Initial image features were extracted by performing a discrete curvelet transform on each of the training images in each database. The number of scales in the curvelet transform can be determined as follows:

$$Scale = \log_2(\min(I_1, I_2)) - 3, \quad (4.23)$$

where  $I_1$  and  $I_2$  are the length- and width of any image respectively. Therefore, an image of size  $80 \times 80$  is decomposed at 4 scale values (coarse, fine, further fine, and finest) using Equation 4.22 as shown in Figure 4.8. The value or size of angles in each scale or sub-band should be multiples of 4. The coarse (first) and the finest (last) curvelet sub-bands each consist of one coefficient, so angle values do not need to be set. The angle value in the second- and third- fine sub-bands is set at 8 and 16 angular orientations respectively. Sub-band selection is performed which corresponds to the highest value of variation. The amount of variation in each coefficient in all sub-bands is computed for the three databases. Tables 4.1-4.4 show the amount of variation of the sub-bands in the images in all datasets. The directional coefficients in each curvelet scale are symmetrical, as shown in Tables 4.1-4.4 and this justifies the method of selecting coefficients used. Therefore, only the coarse coefficient with size  $(13 \times 13)$ , and four coefficients from the second sub-bands have been selected: (two of size  $12 \times 27$ , and two of size  $27 \times 12$ ).

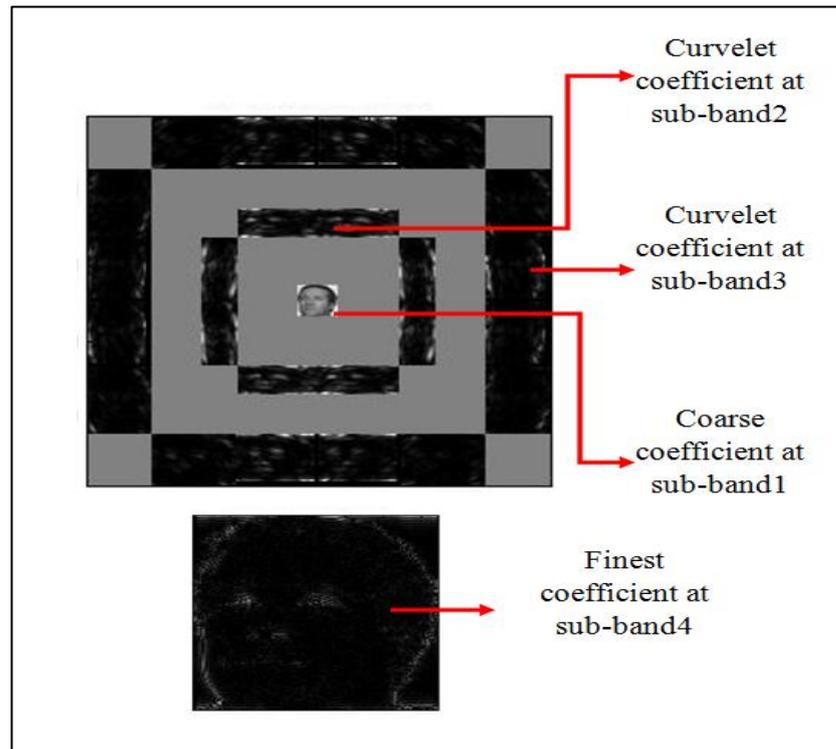


Figure 4.8: Curvelet coefficients at 4 scales and different directions.

Table 4.1: Variance of curvelet sub-band 1 in different databases

Database/ Coefficients	Size (Row Column)	ORL Variance	AR Variance	FERET Variance
<b>Coefficient</b>	13 13	12915	63509	64046

Table 4.2: Variance of curvelet sub-band 2 in different databases

Database/ Coefficients	Size (Row Column)	ORL Variance	AR Variance	FERET Variance
<b>Coefficient_1</b>	12 27	450.56	928.72	794.83
<b>Coefficient_2</b>	12 27	601.77	678.89	800.28
<b>Coefficient_3</b>	27 12	449.36	755.39	781.32
<b>Coefficient_4</b>	27 12	464.43	591.12	922.66
<b>Coefficient_5</b>	12 27	450.56	928.72	794.83
<b>Coefficient_6</b>	12 27	601.77	678.89	800.28
<b>Coefficient_7</b>	27 12	449.36	755.39	781.32
<b>Coefficient_8</b>	27 12	464.43	591.12	922.66

Table 4.3: Variance of curvelet sub-band 3 in different databases

Database/ Coefficients	Size (Row Column)	ORL Variance	AR Variance	FERET Variance
<b>Coefficient_1</b>	22 28	21.31	21.68	28.75
<b>Coefficient_2</b>	20 26	102.17	282.72	282.28
<b>Coefficient_3</b>	20 26	174.37	273.14	333.18
<b>Coefficient_4</b>	22 28	37.30	22.02	36.81
<b>Coefficient_5</b>	28 22	24.84	53.13	52.70
<b>Coefficient_6</b>	26 20	87.07	132.27	249.06
<b>Coefficient_7</b>	26 20	86.93	107.81	130.48
<b>Coefficient_8</b>	28 22	19.57	16.92	57.79
<b>Coefficient_9</b>	22 28	21.31	21.68	28.75
<b>Coefficient_10</b>	20 26	102.17	282.72	282.28
<b>Coefficient_11</b>	20 26	174.37	273.14	333.18
<b>Coefficient_12</b>	22 28	21.314	22.02	36.81
<b>Coefficient_13</b>	20 26	102.17	53.13	52.70
<b>Coefficient_14</b>	20 26	174.37	132.27	249.06
<b>Coefficient_15</b>	22 28	37.30	107.81	130.48
<b>Coefficient_16</b>	28 22	24.84	16.92	57.79

Table 4.4: Variance of curvelet sub-band 4 in different databases

Database/ Coefficients	Size (Row Column)	ORL Variance	AR Variance	FERET Variance
<b>Coefficient</b>	80 80	32.93	46.72	76.48

Tables 4.5 - Table 4.7 show that the proposed sub-band selection method which combines sub-band 1 and four coefficients from sub-band 2 achieves at improvement in error rate compared with that for single subband 1, sub-band 2, sub-band 3, and sub-band 4 in for all three databases ORL, AR, and FERET.

Table 4.5: Error rate (%) in each sub-band for the ORL database.

Sub-band	Error rate (%)
Sub-band1	4.0
Sub-band2	4.9
Sub-band3	6.3
Sub-band4	9.3
<b>Sub-band1+Sub-band2</b> <b>Proposed C-B2DNPP method</b>	<b>3.5</b>

Table 4.6: Error rate (%) in each sub-band for the AR database.

Sub-band	Error rate (%)
Sub-band1	2.1
Sub-band2	3.0
Sub-band3	5.6
Sub-band4	8.4
<b>Sub-band1+Sub-band2</b> <b>Proposed C-B2DNPP method</b>	<b>2.0</b>

Table 4.7: Error rate (%) in each sub-band for the FERET database.

Sub-band	Error rate (%)
Sub-band1	15.4
Sub-band2	16.6
Sub-band3	18.7
Sub-band4	21.4
<b>Sub-band1+Sub-band2</b> <b>Proposed C-B2DNPP method</b>	<b>14.2</b>

After that, three distinctive intermediate feature subsets are obtained by applying the B2DNPP to the selected curvelet coefficients of three different sizes  $(13 \times 13)$ ,  $(12 \times 27)$ , and  $(27 \times 12)$ . These extracted image features preserve the local and spatial

information of the data space, which is important in pattern recognition [32]. The reduction of dimensionality of each feature subset is performed by calculating the weight matrix, then finding two transformation matrices  $U_1$  and  $U_2$  that reduce the dimension of the rows and columns respectively. Therefore, an image of size  $I_1 \times I_2$  in a sub-set is transformed into a feature image of size  $P_1 \times P_2$ , where  $(P_1 < I_1)$  and  $(P_2 < I_2)$ , as explained in Section 4.3.2.2. The values of  $P_1$  and  $P_2$  are set by choosing the best number of eigenvectors according to the overall average error rate for the C-B2DNPP method. In each sub-set, each image feature is converted into a vector. Then, all of the three vectors are combined into one fusion vector, which will be used later in the classification stage. Finally, the fusion vectors extracted from the training and testing images for each dataset are fed to the ELM classifier in order to judge the performance of the C-B2DNPP approach. All of the experiments were undertaken using a PENTIUM 4 PC with 3.2 GHz CPU and 4 GB memory, and MATLAB [169] was used to carry out all of these experiments. Each experiment was repeated 50 times, and the average error rate taken. Different tests were performed on each database using a certain number training samples per subject in each different experiment. The remaining images per subject were used for the testing phase. For  $N_S$  facial images per subject, and  $N_{T_i}$  training images for each subject, there are  $N_W$  various ways to select these  $N_{T_i}$ . The  $N_W$  can be calculated as follows:

$$N_W = \frac{N_S!}{(N_S - N_{T_i})! \times N_{T_i}!} \quad (4.24)$$

For example, for ORL databases, we have  $N_S=10$  and  $N_{T_i}=5$ , therefore  $N_W=252$ . In order to cover significant portion of the space, we performed the experiments for 50 times. The best number of eigenvectors was based on performance, according to the overall average error rate. The B2DNPP is based on unsupervised setting and so does not need any further information to determine the value of  $k$  nearest neighbor images.

### 4.6.2 Experiment using ORL synthetic data

This experiment on the ORL database was performed using 40 subjects for both training and test phases. Five images per subject were randomly selected for the training phase and the remaining 5 images for testing. Table 4.8 illustrates the best error rate from all of the methods used in the experiments. From this table, it is clear that the B2DNPP and C-B2DNPP achieve the best error rates compared with the other methods, including the 2DNPP [50] and curvelet-B2DPCA [54]. The methods proposed here give decreases in error rate of more than 40% and 32% respectively compared to 2DNPP and curvelet-B2DPCA.

Figure 4.9 and Figure 4.10 show the comparison between the proposed C-B2DNPP method and the two-dimensional methods 2DPCA, 2DLDA, 2DLPP, 2DNPP, as well as their bidirectional algorithm forms (2D)2PCA, (2D)2FLD, B2DLPP, C-B2PAC. In this experiment, the error rate (%) is computed for various numbers of training images for each individual. A different number  $T_n$  ( $T_n=2, 3, 4, 5, 6$ ) of images per subject was randomly selected for the training set, and the rest of the images were used as testing images.

Table 4.8: Comparison of error rates (%) using the ORL database.

Method	$p_1$	$p_2$	Error rate (%)
2DPCA	5	64	10.5
2DLDA	5	64	6.4
2DLPP	5	64	6.8
<i>2DNPP</i>	9	64	5.9
(2D)2DPCA	15	20	8.5
(2D)2FLD	14	10	4.9
B2DLPP	11	13	5.5
<i>C-B2DPCA</i>	28		5.2
<b>B2DNPP</b>	<b>13</b>	<b>13</b>	<b>4.1</b>
<b>C-B2DNPP</b>	<b>22</b>		<b>3.5</b>

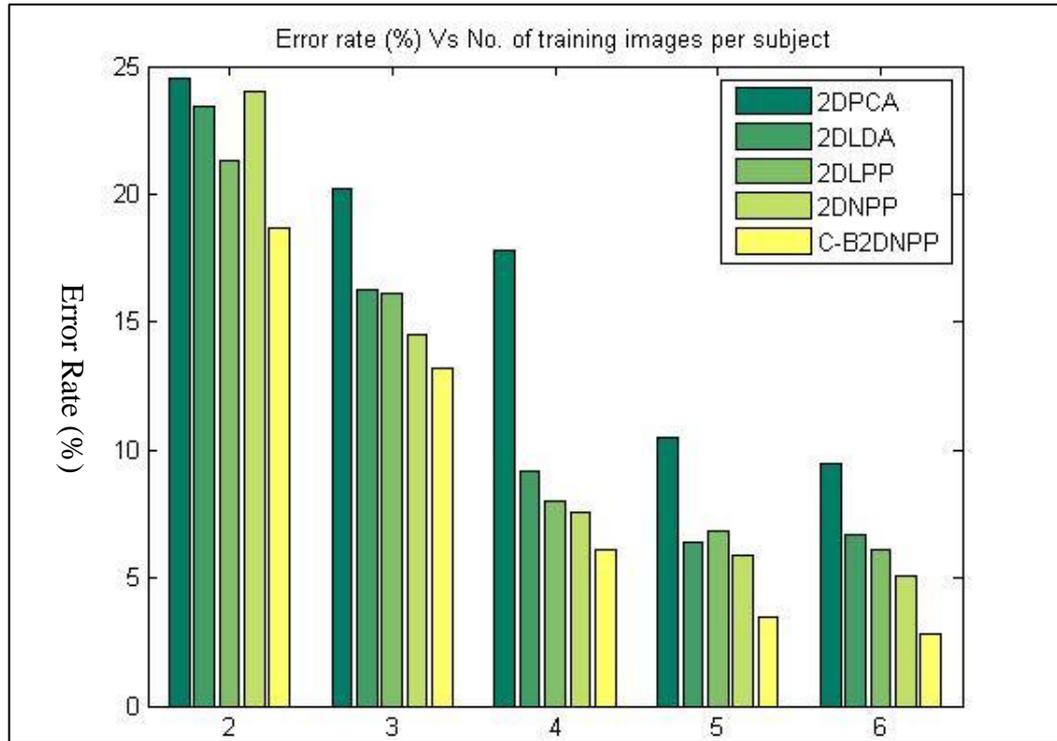


Figure 4.9: Error rate (%) performance of C-B2DNPP and two dimensional methods for ORL database

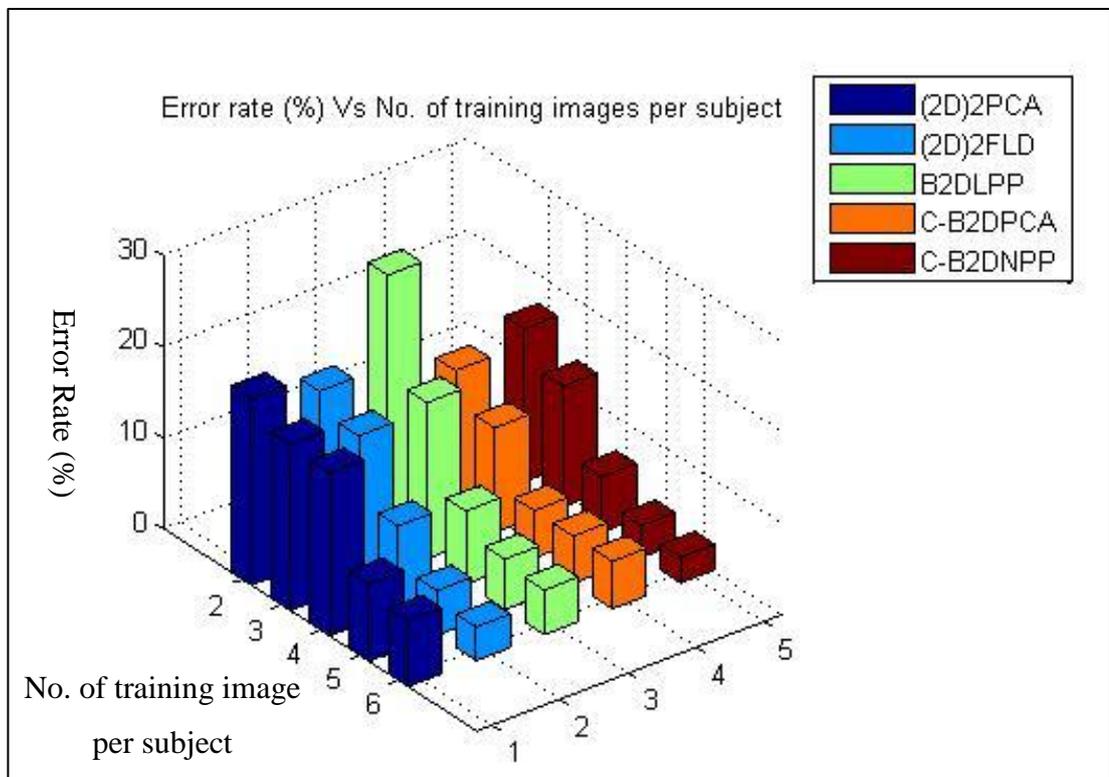


Figure 4.10: Error rate (%) performance of C-B2DNPP and bidirectional two-dimensional methods for ORL database

### 4.6.3 Experiment using AR synthetic data

For the AR database experiments, 50 male and 50 female were used, giving 100 individual subjects in total. Four images from each subject were used, as shown in Figure 4.11 (a)-(d). So that two randomly chosen images were selected for training and the two remaining images for testing.

Table 4.9 shows that C-B2DNPP obtained better results compared with most of the other methods, about 45% and 47% respectively better than 2DNPP and curvelet-B2DPCA. Nevertheless, (2D)2FLD performed slightly better than B2DNPP and C-B2DNPP.

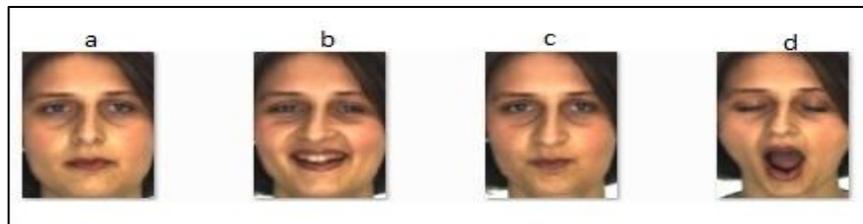


Figure 4.11: A sample from the AR dataset.

Table 4.9: Comparison of error rates (%) using the AR database.

Method	$p_1^{n=2}$	$p_2^{n=2}$	Error rate (%)
2DPCA	10	64	8.2
2DLDA	6	64	2.2
2DLPP	5	64	6.8
2DNPP	5	64	3.7
(2D)2DPCA	15	25	6.5
(2D)2FLD	10	20	1.8
B2DLPP	12	13	4.7
C-B2DPCA	45		3.8
<b>B2DNPP</b>	<b>9</b>	<b>20</b>	<b>2.2</b>
<b>C-B2DNPP</b>	<b>27</b>		<b>2.0</b>

### 4.6.4 Experiment using FERET synthetic data

For the FERET experiments, 100 subjects that who have at least 11 images with different poses in the database were considered. Five training images per class were randomly selected, and the other samples were used for testing. Table 4.10 displays the

results and it is clear that the C-B2DNPP achieved better performance with smaller error rates than all of the other methods, with 27% and 31% performance improvement over 2DNPP and curvelet-B2DPCA respectively.

Table 4.10: Comparison of error rates (%) using the FERET database.

Method	$p_1$	$p_2$	Error rate (%)
2DPCA	13	64	41.8
2DLDA	3	64	26.4
2DLPP	5	64	26.5
2DNPP	9	64	19.7
(2D)2DPCA	15	25	30.1
(2D)2FLD	12	15	19.3
B2DLPP	11	16	17.2
C-B2DPCA	30		20.7
<b>B2DNPP</b>	<b>14</b>	<b>4</b>	<b>16.0</b>
<b>C-B2DNPP</b>	<b>24</b>		<b>14.2</b>

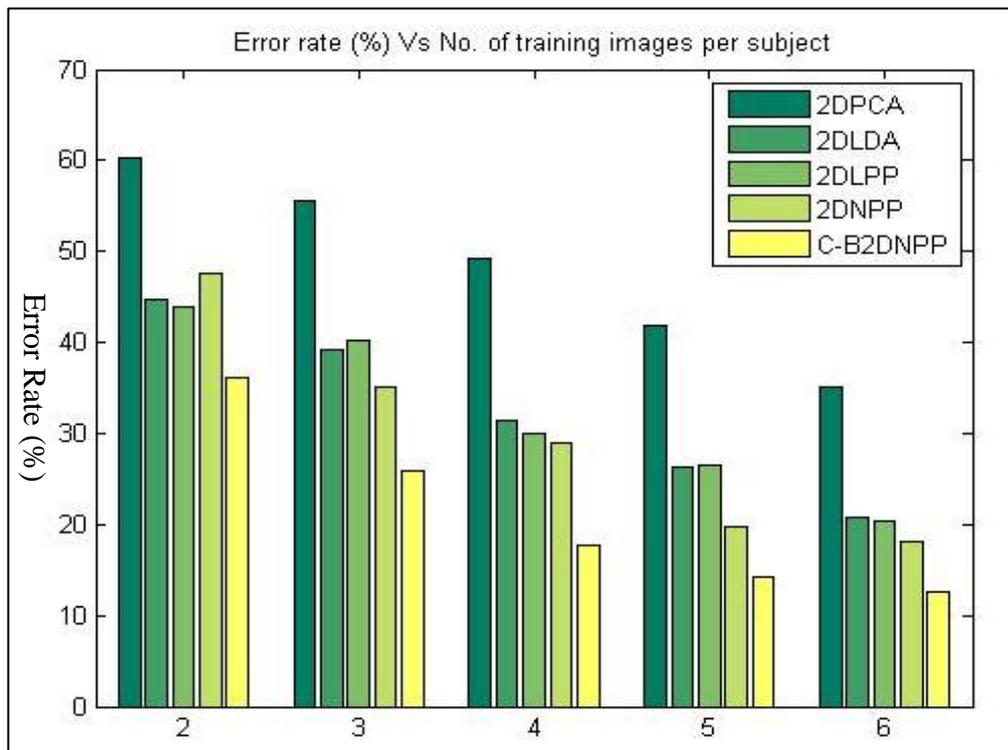


Figure 4.12: Error rate (%) performance of C-B2DNPP and two dimensional methods for FERET database

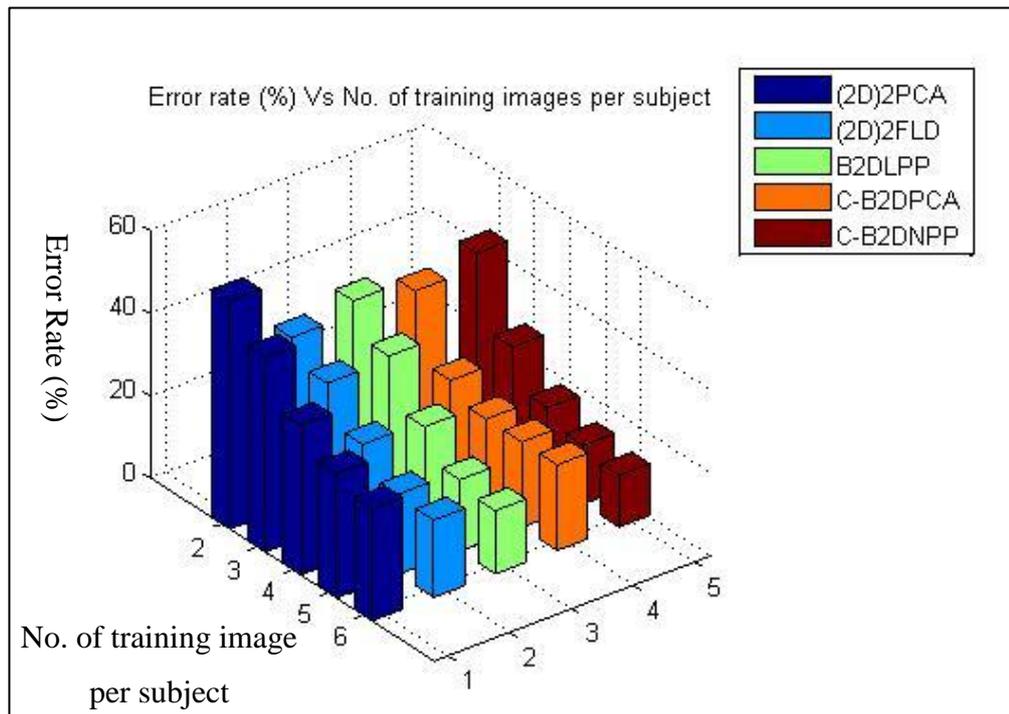


Figure 4.13: Error rate (%) performance of C-B2DNPP and bidirectional two-dimensional methods for FERET database

The results obtained with different numbers of training images are shown in Figure 4.12 and Figure 4.13, which allow a comparison of the proposed C-B2DNPP method and different two-dimensional methods (2DPCA, 2DLDA, 2DLPP, 2DNPP), and the corresponding bidirectional algorithms ((2D)2PCA, (2D)2FLD, B2DLPP, C-B2PAC).

#### 4.6.5 Dimensionality selection

In the C-B2DNPP approach, two targeted dimensionality values ( $P_1, P_2$ ) have to be determined manually before solving the B2DNPP. Therefore, a number of experiments were performed to determine the importance of selecting the values of these two parameters. These experiments were performed for the proposed C-B2DNPP method and all other bidirectional two-dimensional methods mentioned in this chapter ((2D)2PCA, (2D)2FLD, B2DLPP, C-B2PAC) using all three databases (ORL, AR, and FERET). The tests were intended to find the error rate (%) for each of these methods while varying the value of  $(P_1 + P_2)$  from 2 to 100. Started with the value 2, then increasing this value by 2 each time until the value (40), after that the increasing value

will be changed to be (10) until the value (100) . After that, the number of eigenvectors ( $P_1, P_2$ ) was selected for each dataset which lead to the greatest reduction in error rate.

Figure 4.14 shows that both the C-B2DNPP and C-B2DPCA methods work well even when the number of eigenvectors selected for the ORL database is varied. The proposed method performed slightly better than C-B2DPCA. It achieved an error rate of 2.2 using 22 eigenvectors, while C-B2DPCA used 28 features in achieving an error rate of 3.8. For the AR database, Figure 4.15 shows that C-B2DNPP obtained an error rate of 2.0 using 27 features, while C-B2DPCA achieved a 3.8 error rate using 45 eigenvectors. It is clear that the proposed method results in lower error rates than C-B2DPCA, while needing fewer coefficients. Figure 4.16 illustrates that C-B2DNPP achieved the best error rates irrespective of number of features. And it also obtained a lowest error rate of 14.2, which is less with than the other methods even a slightly smaller number of eigenvectors.

These experiments prove that C-B2DNPP obtained better error rates while using the fewest extracted eigenvectors compared to C-B2DPCA. Therefore, the proposed method significantly outperforms C-B2DPCA when used with all the different databases.

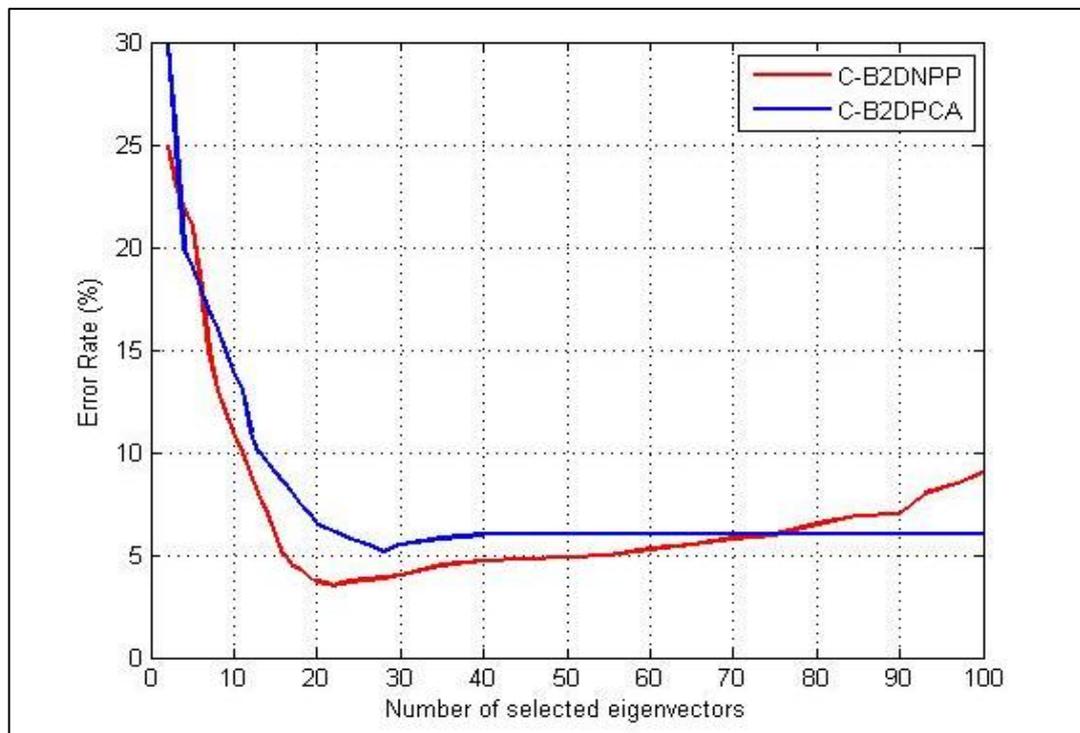


Figure 4.14: Error rate (%) against number of eigenvectors for ORL database.

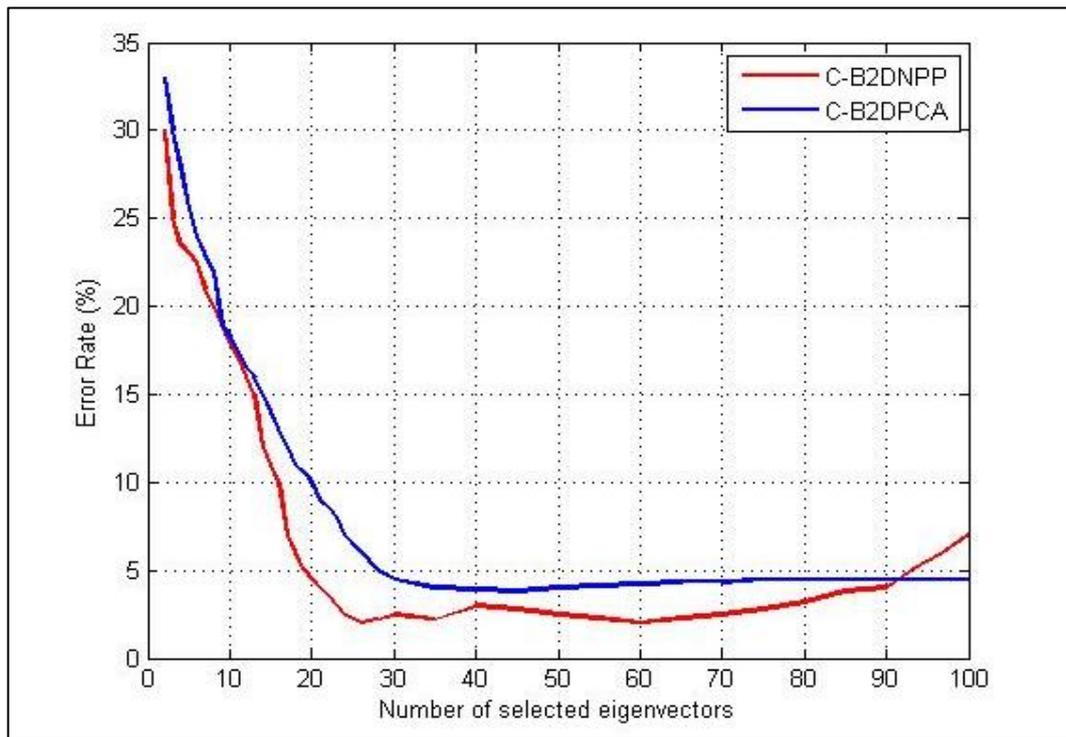


Figure 4.15: Error rate (%) against number of eigenvectors for AR database.

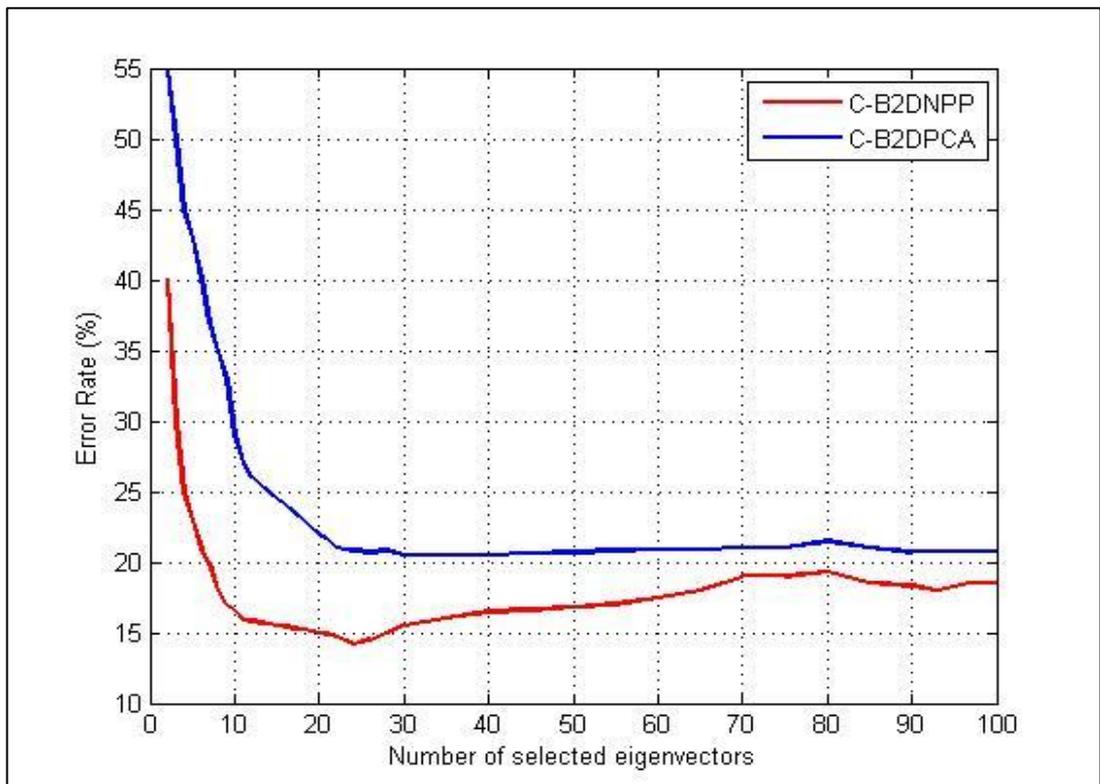


Figure 4.16: Error rate (%) against number of eigenvectors for FERET database.

#### 4.6.6 Discussion

A number of conclusions can be drawn from the results shown in the tables and presented in figures this chapter, as follows:

1. The extensive experiments performed on facial images to evaluate the proposed method have proved that the proposed C-B2DNPP gives promising performance which is better than all of the other method tested. Compared with 2DNPP, the proposed C-B2DNPP method achieves improvements in error rate of more than 45%, 47%, and 27% respectively with the ORL, AR, and FERET databases. Measurement of the numbers of eigenvectors needed to achieve the smallest error rates are reduced by over 88%, 43%, and 88% respectively for the ORL, AR, and FERET databases.
2. Figures 4.9 and 4.12 show from the error rates achieved that the proposed C-B2DNPP approach consistently outperforms all two-dimensional methods (2DPCA, 2DLDA, 2DLPP, 2DNPP) across the ORL and FERET databases. It was not possible to perform the same experiment using the AR database since only four AR images per subject are used in these experiments. The comparisons with bidirectional two-dimensional methods such as (2D)2PCA, (2D)2FLD, B2DLPP, and C-B2PAC, Figures 4.10 and 4.13 showed that the proposed method gives significantly improved recognition performance. Also, it is clear that the B2DPCA-based curvelet transform is suitable when a smaller number of training images is available because it obtained error rates lower than those from C-B2DNPP with two and also three training images per subject. However, C-B2DNPP can achieve much higher classification rates when the number of training images per person increases to more than 3 images.
3. Figures 4.10 and 4.13 show that the number of eigenvectors required for C-B2DNPP to achieve smaller lower error rates decreases when the number of training images per subject increases. The error rate achieved reaches its lowest value using fewer features, and the rate of recognition error gradually increases with increasing numbers of eigenvectors, as illustrated in Figures 4.14– 4.16. It can be noticed from these three figures, however that the classification error rate using the B2DPCA-based curvelet transform not affected by increasing numbers of eigenvectors since lower error rates were achieved. Meanwhile with C-B2DNPP, increasing the number of eigenvectors may slightly increase the error

rate, especially for the ORL and AR databases. So with the C-B2DNPP, it is important to select suitable and correctly targeted values of dimensionality ( $P_1, P_2$ ) and sufficient enough numbers of training images in order to achieved smaller error rates.

4. Compared with the previously study that used only the coarse curvelet coefficient [54], this study selected the coarse sub-band and only half of the coefficients in sub-band 2 associated with the curvelet coefficients that contained high levels of variance, as shown in Tables 4.1-4.4. Therefore, the curvelet selection method applied in this research leads to improvement the performance of the classification system, as shown in Table 4.5- 4.7 for the different databases.
5. The B2DPCA-based curvelet transform method proposed before [54] calculated two weight matrices which independently reduced the column and row directions. The authors of that study found a weight matrix in order to find the transform matrix that reduce of the column direction, and then calculated another weight matrix to extract the projection matrix that reduced the rows. Unlike with B2DPCA, the B2DNPP finds one weight matrix which is used to obtain two transformation matrices that immediately translate the rows and columns of the image data from high to low dimensions using Equation 4.13. This method helps further reduce the computational cost of the dimensionality reduction process.
6. Figures 4.14- 4.16 show that the proposed C-B2DNPP is better than C-B2DPCA for the ORL and AR, and even the FERET datasets. It achieved lower error rates with fewer eigenvectors. This has proven that adding four coefficients from the second curvelet sub-band can improve the classification rate and reduce the number of the features needed to achieve better error rates.
7. Although the proposed B2DNPP worked well compared with most other methods for all of the databases used, and the C-B2DNPP further improved the performance of classification,
8. Table 4.9 shows that (2D)2FLD obtained a slightly better error rate only for the AR database with images which contain variations in facial expression, achieving a recognition error rate 0.2% higher than that of C-B2DNPP. The conclusion from this case is that global information and variation could be important in increasing recognition rate when classifying facial images with which contain different expressions for the same subject.

## 4.7 Summary

This chapter has introduced two new dimensionality reduction approaches. The first is called B2DNPP and addresses the problem of bidirectional two-dimensional human facial recognition. It expands the two-dimensional neighborhood preserving projection (2DNPP) method to its bidirectional two-dimensional form. Compared with 2DNPP, which performs the reduction only on one direction in an image, the proposed B2DNPP works on the two directions of image rows and columns. So, it has the ability to reveal the variations between these two directions. The second approach presented in this chapter is called C-B2DNPP and is a new feature extraction method based on the proposed B2DNPP algorithm which uses feature coefficient sets obtained by the curvelet transform. The proposed method uses a new method of selecting sub-bands depending on the levels of variation in the curvelet coefficients. Therefore, it combines sub-band 1 and four coefficients from sub-band 2 and this improves the error rates achieved compared with those from using a single subband 1, sub-band 2, sub-band 3, and sub-band 4 in all three databases; ORL, AR, and FERET. An extreme learning machine (ELM) for single-hidden layer feedforward neural networks was used in order to examine the performance of the C-B2DNPP framework.

To evaluate the performance of B2DNPE and C-B2DNPP, experiments have been performed on the three benchmark human facial databases, ORL, AR, and FERET. These experiments show that the performance of B2DNPE is more accurate than other methods such as 2DPCA, 2DLDA, 2DLPP, 2DNPP, (2D)2PCA, (2D)2FLD, B2DLPP, and C-B2PAC. It outperforms the 2DNPP method in all of the experiments on various databases with respect to recognition rate and the number of selected eigenvectors. Compared with the 2DNPP, which achieved classification error rates of 5.9%, 3.7%, and 19.7%, the B2DNPP achieved rates of 4.1%, 2.2%, and 16.0 error rate for the ORL, AR, and FERET respectively. Also, the number of eigenvectors used with B2DNPP was reduced over 88%, 43% and 88% for the ORL, AR, and FERET respectively.

The results obtained also show that the features selected by B2DNPP based on the curvelet transform deliver much better performance compared to those with the B2DNPP and other methods such as B2DPCA based on the curvelet transform, while still requiring fewer eigenvectors. Therefore, the classification error rate using C-B2DNPP is further decreased, achieving approximately over 14%, 9% and 11%

improvements in comparison with B2DNPP for the ORL, AR, and FERET databases respectively, whilst number of features is significantly reduced by over 86%, 85% and 57%. So C-B2DNPE outperforms all of the other methods cited in this thesis with respect to the classification error rate, number of eigenvectors, and computational time.

In general, the nature of facial images is a result of the composite consequence of multiple factors. Therefore, it is better to extract useful information from tensorial data directly rather than converting 2-D images into vectors or matrices. Multilinear algebra offers a natural method to handle images that consist of more than two dimensions or an even larger number of multilinear dimensions. Because of this, and in order to improve the performance of the B2DNPP method, the next chapter expands the general NPP method to its multilinear form, which is called multilinear neighborhood preserving projection (MNPP) for face recognition.

# Chapter 5:

## Multilinear Neighborhood Preserving Projection for Face Recognition

---

The facial images are naturally a composite consequence of multiple factors and modes, resulting in so-called tensorial data. These tensorial objects have to be vectorized in order to apply the NPP to this higher-order data of greater than two dimensions. Thus, it is better to extract useful information from tensorial data directly rather than using vectors. Multilinear algebra offers a natural approach to handle images that are the consequence of any number of multilinear factors and to address the difficult problem of disentangling the constituent factors or modes.

In order to improve NPP, which is suitable for use with vectors, and B2DNPP that is used only for matrices, this chapter proposes a novel method of supervised and unsupervised multilinear neighborhood preserving projection (MNPP) for face recognition. Unlike conventional neighborhood preserving projections, the MNPP method operates directly on the tensorial data rather than on vectors or matrices. It aims to solve the problem of tensorial representation for multidimensional feature extraction, classification and recognition. Thus, the proposed MNPP method addresses the pink shaded empty box as shown in Table 3.1 (page 53) under multi-dimensional projection. As opposed to the traditional approaches such as NPP and 2DNPP, as well as B2DNPP, which derives only one or even two subspaces, in the MNPP method multiple interrelated subspaces are obtained by unfolding the tensor over different tensor directions. The number of subspaces derived by MNPP is determined by the order of the tensor space. This approach involving higher order tensors can be used for face recognition and biometrical security classification problems. The performance advantages of the proposed MNPP method over existing techniques are demonstrated using the three benchmark facial datasets ORL, AR, and FERET. The results prove that MNPP outperforms standard approaches across all experiments in terms of error rate.

This chapter begins by explaining the main difference between the proposed MNPP method and some of the other existing approaches, as well as describing the advantages of the MNPP method in Section 5.1. Section 5.2 then formulates the multilinear analysis for the NPP method. Section 5.3 then discusses issues with the MNPP design, including the initialization of projection, the iterative algorithm, the termination procedure, convergence, and the projection order. Section 5.4 demonstrates the associated recognition problem. Section 5 then describes the results of the proposed method using the three facial datasets with a discussion of the results and a summary of the major findings of this work. Finally, the conclusions are provided in Section 5.6.

## **5.1 Introduction**

Face images are naturally in the form of second or higher order tensors, and video sequences with 2-D gray level images can also be viewed as third-order tensors with column, row, and depth modes. In the most active area of biometrics research, and especially in the fields of face recognition and detection, third-order tensors have become an important research area in attempts to reveal the essential structures of data analysis. The traditional linear subspace learning used in dimensionality reduction techniques represents input data as vectors or 2-D matrices and is used to achieve an optimal linear mapping to a lower-dimensional space. Unfortunately, these representations become inadequate when dealing with tensorial data, and they result in the loss of information about the natural structure and correlations in the original multidimensional data. One of the key problems in data analysis for pattern recognition and computer vision is finding a suitable type of representation of the data. It is more appropriate to handle tensor representation directly rather than relying on vector or matrix representations. Thus, multilinear subspace learning which operates directly on the tensor objects is desirable, and such methods have great potential in processing such objects. In the last few years, the multilinear algebra of higher order tensors has received wide attention, as many researchers have begun to represent data in their natural form [45, 46, 170-172]. As such, it has been applied in analyzing the multifactor structure of images [45]. Therefore, a multi-dimensional neighborhood preserving projection method (MNPP) is developed in the present research for multilinear feature extraction, as shown in Figure 5.1. This process consists of three main steps: 1) pre-processing, which normalizes the tensorial data to ensure that all tensors are of the same size [104]; 2) the MNPP framework, which can preserve the relationships among the

natural tensorial or multilinear data as the main element of this system; and 3) classification, which recognizes faces by classifying the MNPP feature vectors.

The advantages of the proposed MNPP method over 1-D and 2-D methods can be summarized in four categories as follows. Firstly, the MNPP method naturally represents multidimensional objects as higher-order tensors. Secondly, the MNPP method maps high-dimensional tensors onto lower dimensional data space by implementing compression in all directions. And thirdly this method preserves the inherent geometry of data samples by eliminating noise and redundant data. Finally, the MNPP method supports robust recognition despite the presence of many different complicating factors such as changes in illumination. Unlike 2DNPP [50], which implements compression and preserves relationships in only the row direction, MNPP performs reduction and preserves the hidden relationships in all directions. Furthermore, it has other significant advantages over some of the other multilinear face recognition methods, such as MPCA [46], DATER [26] and ND-TSNE [44]. In comparison with MPCA and DATER, which both save the global structure of the data samples at the cost of losing major information relevant to the local geometry of the neighbors, MNPP protects the local manifold structure, which is more important than the global Euclidean structure. This is especially relevant in face recognition and data reduction problems. A categorization showing whether or not the different methods can operate in either supervised or unsupervised learning modes is shown in Table 5.1. Unlike ND-TSNE, which is sensitive to the selected width of a Gaussian envelope, MNPP does not require any selection of parameters during the construction of the neighborhood weighting affinity matrix. Therefore, using the suggested approach, MNPP can enhance view-based facial recognition and provides levels of recognition performance better than all of the above-mentioned methods.

The proposed method has been effectively tested on the three face datasets ORL [117], AR [118] and FERET [119, 120] of which each contains images with varying parameters of viewpoint, illumination and facial expression. The experiments show that MNPP obtains significantly improved results and avoids the significant problems of other 1-D, 2-D and competing multi-dimensional methods [14, 15, 31, 33, 46-50, 53, 171, 172]. Also, it is clear that the proposed method can be especially useful when applied in of human face recognition, where database samples are naturally represented

as matrices or higher-order tensors instead of vectors. It has been seen that superior results can be obtained with the MNPP in comparison to the 2D NPP [50].

This chapter offers a number of research contributions. Firstly, it proposes a new method of multilinear neighborhood preserving projection (MNPP), which naturally extends the standard 1-D NPP to the multilinear case. Beside this, MNPP can be implemented in either an unsupervised or supervised way, where the latter may require prior knowledge related to the number of neighbors. Secondly, the MNPP preserves and naturally estimates the intrinsic local geometric and topological properties of the tensorial data space. Thirdly, it solves the problem of tensors representation in tensor object feature extraction and recognition. Fourthly, it does not suffer from the so-called curse of dimensionality the associated high computational costs. Finally in this chapter, broad comparisons are made between the MNPP and other methods, in particular with the 2DNPP and other techniques such as MPCA, DATER, ND-TSNE, 2DNPP, 2DPCA, 2DLDA, 2DLPP, PCA, LDA, NPP, and LPP.

Table 5.1: Categorization of Different Methods

<b>Method</b>	<b>Supervised</b>	<b>Unsupervised</b>
MNPP	Y	Y
2DNPP[50]	Y	Y
DATER [26]	Y <sup>1</sup>	N
ND-TSNE [44]	Y <sup>1</sup>	N
MPCA [46]	N	Y

<sup>1</sup> require additional information about the class labels

## **5.2 Multi-dimensional Neighborhood Preserving Projection Approach (MNPP)**

As previously highlighted, the number of data processing tasks required in manipulating tensorial data has increased enormously over recent years so as to become a serious limitation. In order to process this tensorial data and its natural multi-dimensional structure, it is necessary to reduce its dimensionality or use feature extraction techniques.

This section proposes a multilinear neighborhood preserving projection recognition (MNPP) method for feature extraction, which can learn an optimal subspace that preserves the local relationships of the data manifold and is also able to obtain an acceptable recognition rate in facial recognition. A weight matrix, which describes the relationship between each data point and its  $k$  nearest neighbors ( $k$ -NNs) in an optimal way, is built according to [14] and as shown in Figure 5.1. In order to preserve the data space with reduced dimensionality, an optimal projection of the neighborhood structure must be found. As previously indicated in Table 1, the MNPP can be performed in either supervised or unsupervised mode. When class information is available, this can be incorporated so as to build a better weight matrix. Further detailed theoretical justifications of the MNPP algorithm are provided in section 3.3.

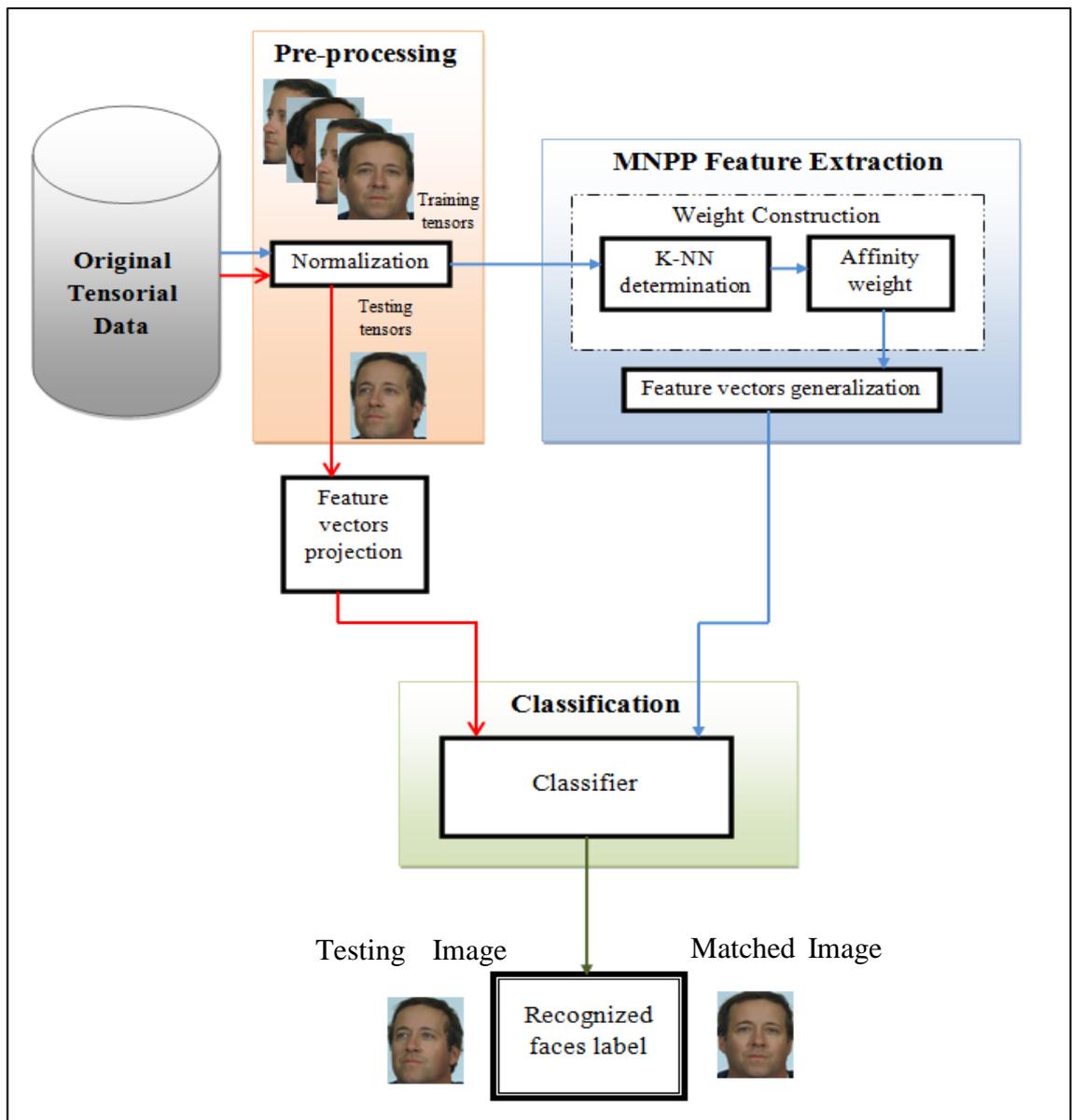


Figure 5.1: Block Diagram of MNPP System for Face Recognition

### 5.2.1 Multilinear dimensionality reduction problem formulation

To explain the general problem of multilinear dimensionality reduction, let  $\{\mathcal{X}\}$  be a set of  $M$  training tensor objects ( $\mathcal{X}_m \in R^{I_1 \times I_2 \times \dots \times I_N}$ ,  $m = 1, 2, \dots, M$ ). The objective of the MNPP is to define  $N$  multilinear projection matrices  $\{\mathbf{U}^{(n)} \in R^{I_n \times P_n}$ ,  $n = 1, 2, \dots, N\}$ , which maps the original  $M$  tensor  $\mathcal{X}_m \in R^{I_1 \times I_2 \times \dots \times I_N}$  into a smaller tensorial set  $\mathcal{Y}_m \in R^{P_1 \times P_2 \times \dots \times P_N}$  with  $P_n < I_n$ ,  $n = 1, 2, \dots, N$ , such that each tensor object  $\mathcal{X}_m$  can be represented directly by  $\mathcal{Y}_m$  using the transformation function  $\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}^{(1)T} \times_2 \mathbf{U}^{(2)T} \times \dots \times_N \mathbf{U}^{(N)T}$ . The new dimensionality  $P_n$  is assumed to be known for all modes  $n$ ,  $n = 1, 2, \dots, N$ . The objective of MNPP is to determine  $N$  transformation matrices  $\{\mathbf{U}^{(n)} \in R^{I_n \times P_n}$ ,  $n = 1, 2, \dots, N\}$  that minimize the cost function  $\phi(\mathcal{Y})$ , can be shown as:

$$\{\mathbf{U}^{(n)}, n = 1, 2, \dots, N\} = \arg \min_{\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \dots, \mathbf{U}^{(N)}} \phi(\mathcal{Y}) . \quad (5.1)$$

### 5.2.2 Neighborhood weighting

We can construct a nearest neighbor graph  $\mathcal{G}$  with  $M$  tensor objects in order to model the local geometrical structure of the data manifold [173]. Let  $\mathbf{W} \in R^{M \times K}$  be the optimal weight matrix of  $\mathcal{G}$  where the data reflects intrinsic geometric properties and relates each tensor object to its neighbors in a local way. Therefore, each component of  $w_{ij}$  is the optimal weight of the edge between the tensor  $i$  and tensor  $j$ . The basic assumption here is that each tensor sample and its  $k$ -nearest neighbor points lie on a locally linear manifold. Therefore, each tensor object can be rebuilt using the linear combination of its  $k$ -nearest neighbor [31]. Here, the number of neighbors  $k$  is assumed to be predetermined. In preserving the local structure, a proper criterion of construction of the affinity matrix  $\mathbf{W}$  minimizes the following objective function:

$$\delta(\mathbf{W}) = \sum_i \left\| \mathcal{X}_i - \sum_j w_{ij} \mathcal{X}_j \right\|^2 , \quad (5.2)$$

with the constraint that  $\sum_j w_{ij} = 1, j = 1, 2, \dots, k$ . The objective function in Equation 5.2 increases the error rate if  $\mathcal{X}_i$  is mapped too far from its neighbor point  $\mathcal{X}_j$ . Therefore, minimizing this distance ensures that  $\mathcal{X}_i$  and  $\mathcal{X}_j$  are close to each other. The components of  $\mathbf{W}$  can be defined by [33, 72]:

$$w_{i,:} = \frac{\mathbf{G}^{-1}\mathbf{e}}{\mathbf{e}^T \mathbf{G}^{-1}\mathbf{e}}, \quad (5.3)$$

where  $\mathbf{e}$  is the vector such that,  $\mathbf{e} = [1, \dots, 1]^T$ , and  $\mathbf{G} \in R^{K \times K}$  denotes the local Gramian matrix associated with the column vector  $s_i \in R^{I_1 I_2 \dots I_N \times 1}$  of tensor  $\mathcal{X}_i$ . The entries of  $\mathbf{G}$  can be determined by:

$$g_{uv} = (s_i - s_u)^T (s_i - s_v) \in R^{K \times K}. \quad (5.4)$$

The above is a constrained least squares problem that can be solved by Equation 5.3 using the inverse of  $\mathbf{G}$ . The MNPP method uses the same affinity weight matrix as reported in a previous study [33]. This matrix is used to reconstruct  $\mathcal{X}_i$ , which is the data point that is derived from its neighbors in the original space. The weights  $w_{ij}$  are invariant to rotation, scaling and translation [72]. As a result, they preserve the intrinsic geometry of the underlying manifold. The derivation of the projection  $\mathcal{Y}_i$  in the lower dimensional space is then produced by its corresponding neighbors.

### 5.2.3 The derivation of MNPP solution and projection learning

The MNPP approach decomposes each mode- $n$  of tensor  $i$  with the objective of taking into consideration the neighborhood relationships of the training dataset  $n=1, 2, \dots, N$ . Therefore, it models the  $N$ -dimensional data in a natural way without any rasterization. Such methods extract the essential information from the natural tensorial data by finding a more compact representation of the original data. The proposed method therefore captures the maximum variation of the input tensors with very little loss of information.

Let  $\mathbf{U}^{(n)} \in R^{I_n \times P_n}$  be the mode- $n$  transformation matrices with  $P_n < I_n$ , where  $P_n$  is the dimension of the reduced space  $I_n$  and  $I_n$  is the dimension number  $n$  for

$n = 1, 2, \dots, N$ . Since the projection onto an  $N$  order tensor subspace consists of  $N$  projections to vector subspaces, optimization problems can be solved by finding the  $\mathbf{U}^{(n)}$  which minimizes the reconstruction error in mode- $n$  (for  $n = 1, 2, \dots, N$ ) in an iterative procedure along the pseudo code steps summarized in Figure 5.2.

Let  $\{\mathbf{U}^{(n)}, n = 1, 2, \dots, N\}$  be the solution to Equation 5.1. Using the known matrix  $\mathbf{U}^{(r)}, r = 1, 2, \dots, N, r \neq n$ , the linear projection matrix  $\mathbf{U}^{(n)}$  can be found by minimizing the objective function under the constraint that:

$$\mathbf{U}^{(n)} = \arg \min_{\mathbf{U}^{(n)T} \mathbf{V}^{(n)} \mathbf{U}^{(n)} = \mathbf{I}} \left( \mathbf{U}^{(n)T} (\mathbf{V}^{(n)} - \mathbf{Z}^{(n)}) \mathbf{U}^{(n)} \right) \quad (5.5)$$

where  $\mathbf{V}^{(n)} = \sum_i \mathcal{X}_i^T \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} \mathcal{X}_i \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)}$ ,

and

$$\mathbf{Z}^{(n)} = \left[ \begin{aligned} & \left( \sum_{i,j} \mathcal{X}_j^T \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} w_{ij} \mathcal{X}_i \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} + \left( \sum_{i,j} \mathcal{X}_j^T \prod_{r=1; r \neq n}^N \times_r w_{ij} \mathbf{U}^{(r)} \mathcal{X}_i \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} \right)^T \right) \\ & - \sum_{i,k,l} \mathcal{X}_i^T \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} w_{ik} w_{il} \mathcal{X}_j \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)}. \end{aligned} \right]$$

Following the details of the above derivation:

Let  $\mathbf{W} \in R^{M \times K}$  be the weight matrix. The objective function given in Equation 5.2 can be rewritten as a multilinear tensor projection form as follows:

$$\delta(\mathcal{Y}) = \sum_i \left\| \mathcal{X}_i \prod_{r=1}^N \times_r \mathbf{U}^{(r)} - \sum_j w_{ij} \mathcal{X}_j \prod_{r=1}^N \times_r \mathbf{U}^{(r)} \right\|^2. \quad (5.6)$$

To solve Equation 5.6 with the unknown matrix  $\mathbf{U}^{(n)}$ , the data is first projected using the known matrix  $\mathbf{U}^{(r)}, r = 1, 2, \dots, N, r \neq n$  onto a lower subspace, and then unfold the

projected data is unfolded into a matrix onto the mode- $n$  using the mode- $n$  unfolding matrix. Since  $\|\mathbf{A}\|^2 = tr(\mathbf{A}\mathbf{A}^T)$ , the objective  $\delta(\mathbf{Y})$  of Equation 5.6 can be rewritten as follows:

$$\delta(\mathbf{Y}) = \sum_i \left\{ tr \left[ \left( \mathbf{Y}_i^{(n)} - \sum_j w_{ij} \mathbf{Y}_j^{(n)} \right) \left( \mathbf{Y}_i^{(n)} - \sum_j w_{ij} \mathbf{Y}_j^{(n)} \right)^T \right] \right\}$$

The above Equation can be written using Equation 3.9 (Section 3.2.2, Page 48) as follows:

$$\delta(\mathbf{Y}) = \sum_i \left\{ tr \left[ \left( \mathbf{U}^{(n)T} \mathcal{X}_i \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} - \sum_j w_{ij} \mathbf{U}^{(n)T} \mathcal{X}_j \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} \right) \left( \mathbf{U}^{(n)T} \mathcal{X}_i \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} - \sum_j w_{ij} \mathbf{U}^{(n)T} \mathcal{X}_j \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} \right)^T \right] \right\}$$

Then,

$$\delta(\mathbf{Y}) = \sum_i \left\{ tr \left[ \mathbf{U}^{(n)T} \left( \mathcal{X}_i \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} - \sum_j w_{ij} \mathcal{X}_j \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} \right) \left( \mathcal{X}_i \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} - \sum_j w_{ij} \mathcal{X}_j \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} \right)^T \mathbf{U}^{(n)} \right] \right\}$$

Expanding the brackets,

$$\delta(\mathbf{Y}) = \sum_i \left\{ tr \left[ \mathbf{U}^{(n)T} \left( \mathcal{X}_i^T \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} \mathcal{X}_i \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} - \sum_j \mathcal{X}_j^T \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} w_{ij} \mathcal{X}_i \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} - \sum_j \mathcal{X}_i^T \prod_{r=1; r \neq n}^N \times_r w_{ij} \mathbf{U}^{(r)} \mathcal{X}_j \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} + \sum_{k,l} \mathcal{X}_k^T \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} w_{ik} w_{il} \mathcal{X}_l \prod_{r=1; r \neq n}^N \times_r \mathbf{U}^{(r)} \right) \mathbf{U}^{(n)} \right] \right\}$$

By changing the order of the trace operator and the summation, the pervious equation can be written as follow:

$$\begin{aligned} \delta(\mathcal{Y}) &= tr \left[ \mathbf{U}^{(n)T} \left( \begin{aligned} &\sum_i \mathcal{X}_i^T \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} \mathcal{X}_i \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} - \sum_{i,j} \mathcal{X}_j^T \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} w_{ij} \mathcal{X}_i \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} \\ &- \sum_{i,j} \mathcal{X}_i^T \prod_{r=1; r \neq n}^N w_{ij} \mathbf{U}^{(r)} \mathcal{X}_j \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} + \sum_{i,k,l} \mathcal{X}_i^T \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} w_{ik} w_{il} \mathcal{X}_j \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} \end{aligned} \right) \mathbf{U}^{(n)} \right] \\ &= tr \left[ \mathbf{U}^{(n)T} \left( \begin{aligned} &\sum_i \mathcal{X}_i^T \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} \mathcal{X}_i \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} - (\sum_{i,j} \mathcal{X}_j^T \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} w_{ij} \mathcal{X}_i \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} \\ &+ (\sum_{i,j} \mathcal{X}_j^T \prod_{r=1; r \neq n}^N w_{ij} \mathbf{U}^{(r)} \mathcal{X}_i \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)})^T + \sum_{i,k,l} \mathcal{X}_i^T \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} w_{ik} w_{il} \mathcal{X}_j \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} \end{aligned} \right) \mathbf{U}^{(n)} \right] \end{aligned}$$

Hence, the above leads to:

$$\delta(\mathcal{Y}) = tr \left( \mathbf{U}^{(n)T} (\mathbf{V}^{(n)} - \mathbf{Z}^{(n)}) \mathbf{U}^{(n)} \right), \quad (5.7)$$

where

$$\mathbf{V}^{(n)} = \sum_i \mathcal{X}_i^T \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} \mathcal{X}_i \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)},$$

and

$$\begin{aligned} \mathbf{Z}^{(n)} &= (\sum_{i,j} \mathcal{X}_j^T \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} w_{ij} \mathcal{X}_i \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} \\ &+ (\sum_{i,j} \mathcal{X}_j^T \prod_{r=1; r \neq n}^N w_{ij} \mathbf{U}^{(r)} \mathcal{X}_i \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)})^T) - \sum_{i,k,l} \mathcal{X}_i^T \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)} w_{ik} w_{il} \mathcal{X}_j \prod_{r=1; r \neq n}^N \mathbf{U}^{(r)}. \end{aligned}$$

As the projection onto an  $N$  th-order tensor subspace consists of  $N$  projections to  $N$  vector subspaces, the alternating projection method in the ALS method [113, 114] is adopted for the optimization of all modes. In ALS, the optimal base vectors on mode- $n$  are obtained by fixing the base vectors on the other modes and cycling for the remaining variables. In other words, the optimization subproblems can be solved by finding the

$\mathbf{U}^{(n)}$  in the mode- $n$  vector subspace, conditional on the projection matrices in the other modes. Therefore, it can be seen that, by minimizing the objective function under constraint  $(\mathbf{U}^{(n)T} \mathbf{V}^{(n)} \mathbf{U}^{(n)} = \mathbf{I})$ , the linear projection  $\mathbf{U}^{(n)}$  can be obtained as:

$$\mathbf{U}^{(n)} = \arg \min_{\mathbf{U}^{(n)T} \mathbf{V}^{(n)} \mathbf{U}^{(n)} = \mathbf{I}} \left( \mathbf{U}^{(n)T} (\mathbf{V}^{(n)} - \mathbf{Z}^{(n)}) \mathbf{U}^{(n)} \right) \quad (5.8)$$

The minimization of Equation 5.8 can be efficiently solved by converting it to the generalized eigenvalues problem as follows:

$$(\mathbf{V}^{(n)} - \mathbf{Z}^{(n)}) \mathbf{U}^{(n)} = \lambda \mathbf{V}^{(n)} \mathbf{U}^{(n)}. \quad (5.9)$$

Corresponding to the  $P_n$  smallest eigenvalues,  $\mathbf{U}^{(n)}$  is the optimal transformation matrix that consists of the  $P_n$  eigenvectors, which can be used to project each testing tensor object into an MNPP basis for each mode in order to provide an optimum input to the classifier. The value of  $\lambda$  is the eigenvalues solution to the minimization problem. Figure 5.2 shows the pseudo code of the proposed MNPP method.

#### **5.2.4 Supervised MNPP approach**

Here, the unsupervised version of the MNPP method is described without considering the class information for the training faces. It is easy to extend MNPP to a supervised setting when the labels of training faces are available. This is achieved by selecting the value of  $k$ , which is the number of nearest neighbors associated with the number of training sample  $\mathcal{X}_i$ . In a supervised setting, the nearest neighbors of a data sample  $\mathcal{X}_i$  are only considered if they belong to the same class as  $\mathcal{X}_i$ . Using this setting, the discriminative information can be incorporated into the neighborhood weighting of the MNPP. Thus, the supervised MNPP is able to deliver not only intrinsic geometry information, but also discriminative information that will enable recognition performance to be further improved.

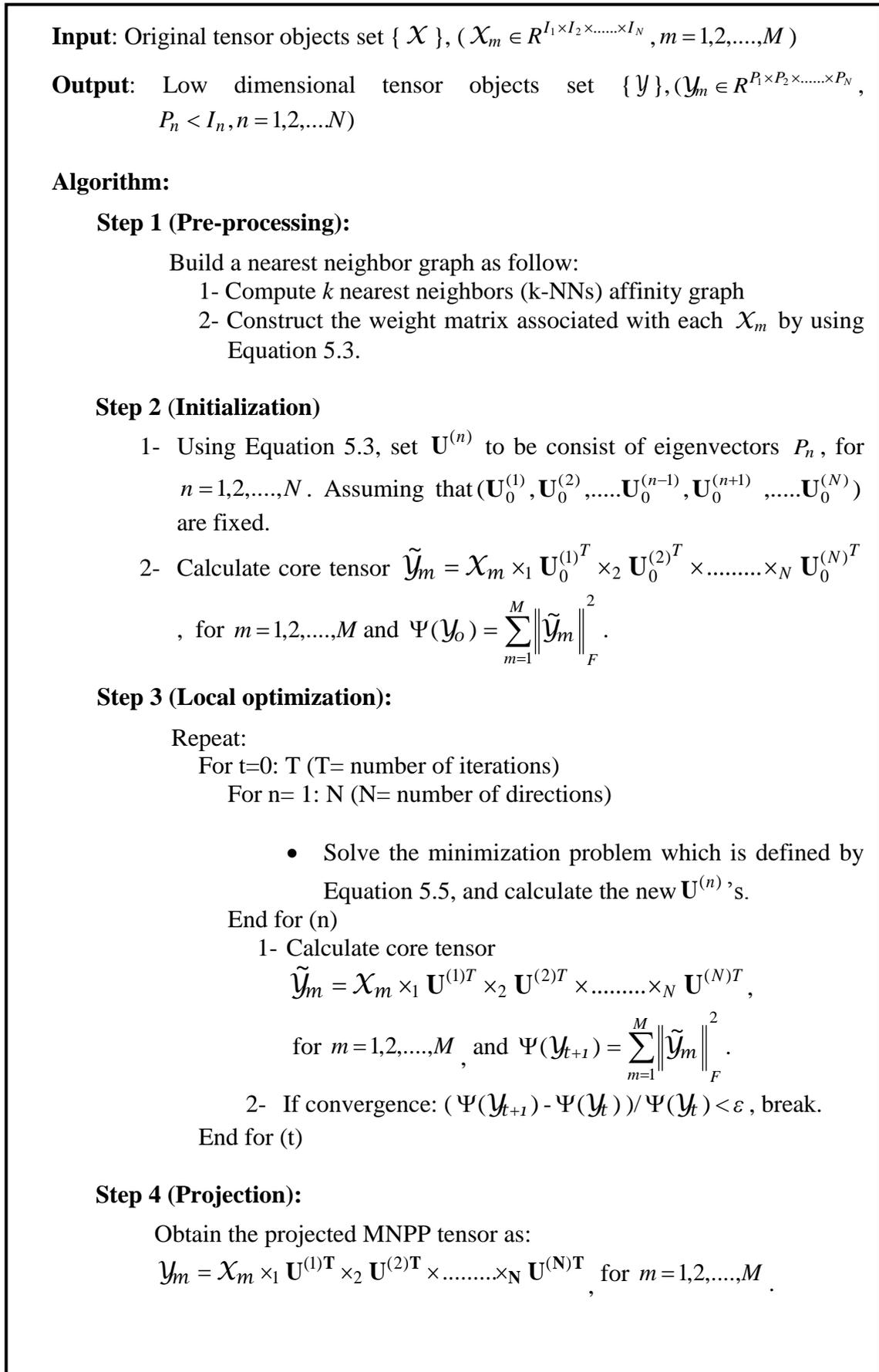


Figure 5.2: The Pseudo Code for the MNPP Proposed Approach

### 5.3 Design of the MNPP Approach

This section discusses some issues relevant to the implementation of the proposed MNPP approach. Firstly, the initial and full projections are illustrated. Next, the structures of the termination and convergences problems are explained. Finally, the projection order for the MNPP is discussed.

#### 5.3.1 Initial projection

A truncation of the full projection is assumed to be a good starting step for any iterative procedure, which increases the speed of the convergence of the iterative solution [46]. As such, all of the first  $P_n$  eigenvectors columns in the projection matrices  $\{\mathbf{U}^{(n)}, n=1,2,\dots,N\}$  are retained. Therefore, to initialise the iterative solution of MNPP, the full multilinear projection matrices  $\{\mathbf{U}^{(n)}\}$  of MNPP are firstly calculated with  $P_n = I_n$  for  $n=1,2,\dots,N$ . Next, all the eigenvalues are arranged in ascending order for each projection matrix. Then, the initial projection matrices  $\{\mathbf{U}_0^{(n)}, n=1,2,\dots,N\}$  are truncated, during which process only the first  $P_n$  eigenvector columns are retained according to the  $P_n$  smallest eigenvalues for  $n=1,2,\dots,N$ , as follows:

#### Initialization:

- Find  $\mathbf{U}_0^{(n)}$  for  $n=1,2,\dots,N$
- Calculate core tensor  $\tilde{\mathbf{Y}}_m = \mathcal{X}_m \times_1 \mathbf{U}_0^{(1)T} \times_2 \mathbf{U}_0^{(2)T} \times \dots \times_N \mathbf{U}_0^{(N)T}$ , for  $m=1,2,\dots,M$
- Calculate  $\Psi(\mathbf{Y}_o) = \sum_{m=1}^M \left\| \tilde{\mathbf{Y}}_m \right\|_F^2$

#### 5.3.2 Iterative algorithm

The truncated projection matrices may yield a good reduction in the tensor dimensions, but it is generally not the optimal solution. Therefore, since the initial projection matrices have been found, each transformation matrix  $\mathbf{U}_0^{(n)}$  for  $n=1,2,\dots,N$  needs to

be recalculated in all other modes. In each mode- $n$  during the iterative procedure, the updated  $\tilde{\mathbf{U}}^{(n)}$  matrix is computed while keeping all other matrices fixed  $\{\mathbf{U}^{(r)}, r = 1, 2, \dots, N, r \neq n\}$ . To this end, the iteration steps in the MNPP approach are described as follows:

**Iteration:**

For  $t=0$ : T (T= number of iterations)

- For  $n=1$ : N

$n=1$

i. Set  $\tilde{\mathbf{U}}_{t+1}^1 = \mathcal{X} \times_2 \mathbf{U}_t^{(2)T} \times \dots \times_N \mathbf{U}_t^{(N)T}$ , for  $m=1, 2, \dots, M$

ii. Flatten tensor  $\tilde{\mathbf{U}}_{t+1}^1$  over mode-1 to find the updated matrix  $\tilde{\mathbf{U}}_{t+1}^1$ .

iii. Set  $\mathbf{U}_{t+1}^1$  to consist of  $P_1$  eigenvector columns according to the  $P_1$  smallest eigenvalues.

$n=2$

i. Set  $\tilde{\mathbf{U}}_{t+1}^2 = \mathcal{X} \times_1 \mathbf{U}_{t+1}^{(1)T} \times_3 \mathbf{U}_t^{(3)T} \dots \times_N \mathbf{U}_t^{(N)T}$ , for  $m=1, 2, \dots, M$

ii. Flatten tensor  $\tilde{\mathbf{U}}_{t+1}^2$  over mode-1 to find the updated matrix  $\tilde{\mathbf{U}}_{t+1}^2$ .

iii. Set  $\mathbf{U}_{t+1}^2$  to consist of  $P_2$  eigenvector columns according to the  $P_2$  smallest eigenvalues.

·  
·  
·

$n=N$

i. Set  $\tilde{\mathbf{U}}_{t+1}^N = \mathcal{X} \times_1 \mathbf{U}_t^{(1)T} \times_2 \mathbf{U}_t^{(2)T} \dots \times_{N-1} \mathbf{U}_t^{(N-1)T}$ , for  $m=1, 2, \dots, M$ .

ii. Flatten tensor  $\tilde{\mathbf{U}}_{t+1}^N$  over mode-1 to find the updated matrix  $\tilde{\mathbf{U}}_{t+1}^N$ .

iii. Set  $\mathbf{U}_{t+1}^N$  to consist of  $P_N$  eigenvector columns according to the  $P_N$  smallest eigenvalues.

End for (n)

- Calculate core tensor  $\mathbf{Y}_m = \mathbf{X}_m \times_1 \mathbf{U}^{(1)\mathbf{T}} \times_2 \mathbf{U}^{(2)\mathbf{T}} \times \dots \times_N \mathbf{U}^{(N)\mathbf{T}}$ , for  $m = 1, 2, \dots, M$
- Calculate  $\Psi(\mathbf{Y}_{t+1}) = \sum_{m=1}^M \left\| \tilde{\mathbf{Y}}_m \right\|_F^2$
- If convergence:  $(\Psi(\mathbf{Y}_{t+1}) - \Psi(\mathbf{Y}_t)) / \Psi(\mathbf{Y}_t) < \varepsilon$ , where  $\varepsilon = 10^{-6}$ , break.

End for (t)

**Output:**  $\{\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times P_n}, n = 1, 2, \dots, N\}$ .

### 5.3.3 Termination

The MNPP method determines termination using the objective function  $\Psi(\mathbf{Y})$ . In other words, the iterative steps halt if there is a small improvement in the objective function. In particular, the iterative process stops if  $(\Psi(\mathbf{Y}_{t+1}) - \Psi(\mathbf{Y}_t)) / \Psi(\mathbf{Y}_t) < \varepsilon$ , where  $\varepsilon$  is a small threshold value defined by the user (here,  $\varepsilon$  is defined to be  $10^{-6}$ ), and  $\Psi(\mathbf{Y}_t)$  as well as  $\Psi(\mathbf{Y}_{t+1})$  are the total objective function in the  $t$  and  $t+1$  iterations, respectively. The  $\Psi(\mathbf{Y})$  in any iteration is defined as follows:

$$\Psi(\mathbf{Y}) = \sum_{m=1}^M \left\| \tilde{\mathbf{Y}}_m \right\|_F^2. \quad (5.10)$$

Also, the iterative algorithm can be terminated easily after T number of iterations in order to reduce the computational cost. In practice, the maximum value of T can be set according to experimental study.

### 5.3.4 Convergence in the MNPP method

In the iterative procedure, the non-negative objective function  $\Psi(\mathbf{Y})$  decreases monotonically (either remaining the same or decreasing). Therefore, the iterative process is shown to be convergent. During the operation of the MNPP iterative algorithm, the projection matrices  $\{\mathbf{U}^{(n)}, n = 1, 2, \dots, N\}$  do converge within a small number of iterations, as indicated in the experimental studies. Thus, the proposed MNPP algorithm quickly converges within 4 iterations for typical tensor samples.

### 5.3.5 Projection order

There is no significant effect of the projection order on the performance of the iteration procedure. In other words, changing the ordering in which the projection matrix is computed does not affect the solution obtained [46]. Therefore, the MNPP uses a sequential order from 1 to  $N$  during its implementation.

### 5.4 MNPP-Based Tensor Object Recognition

The projection coefficients  $\{\mathbf{U}^{(n)}, n = 1, 2, \dots, N\}$  which represent the extracted feature matrices, can be used for facial recognition and classification. This section shows the MNPP recognition framework as shown in Figure 5.1. Since the focus here is on recognition rather than detection, all facial images are manually cropped and aligned with annotated coordinate information about the eyes, and are normalized to ensure that all tensors having the same size. Also, because tensorial data often have different dimensions, therefore all of the input tensors need to be normalized to standard dimensions.

Using the learning projection matrices  $\{\mathbf{U}^{(n)}, n = 1, 2, \dots, N\}$ , the low-dimensional representation of the training tensor sample  $\{\mathcal{X}_m \in R^{I_1 \times I_2 \times \dots \times I_N}, m = 1, 2, \dots, M\}$  can be computed as  $\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}^{(1)T} \times_2 \mathbf{U}^{(2)T} \times \dots \times_N \mathbf{U}^{(N)T}$ . When test tensor data becomes available, its low dimensional representation is first computed as  $\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}^{(1)T} \times_2 \mathbf{U}^{(2)T} \times \dots \times_N \mathbf{U}^{(N)T}$ . The distances between the testing tensor features  $\mathcal{Y}$  and all of the training sample features  $\mathcal{Y}_m$  are then compared, such that its class label is expected to be that of the training sample, where the low-dimensional representation is nearest to  $\mathcal{Y}$  such that:

$$m = \arg \min_m \|\mathcal{Y}_m - \mathcal{Y}\|. \quad (5.11)$$

The test tensor  $\mathcal{Y}$  is then classified as belonging to the  $C_m$  class. In this chapter, the nearest neighbor classifier is used for final classification in all of the experiments, as this is the most widely used distance metric, which also makes the classification easier and quicker [174].

## 5.5 Evaluation of the Performance of MNPP

Some important parameters have to be measured in order to determine the performance of any biometrics system. Any such system identifies a person via identification and verification modes. In identification mode, the person has to be recognized based on distinctive biometric features without any further information being needed. An individual's features are compared with templates stored in the original database in order to derive a set of matching scores. If the scores are less than a threshold value, the classifier identifies the person as a best match in the registration database. The performance of this system can be evaluated using values of correct recognition rate (CRR) and the error rate (ERR), which are defined as follows:

$$CRR = \frac{N_C}{N_T} \times 100, \quad (5.12)$$

$$ERR = 100 - CRR, \quad (5.13)$$

where,  $N_T$  is the total number of images, and  $N_C$  is the total number of correctly accepted images. However, in verification mode the user image is matched against the one template that is claimed when the person uses an additional identity card. Four important parameters are used to evaluate the system's performance: false acceptance rate (FAR) and false rejection rate (FRR), genuine acceptance rate (GAR), and equal error rates (EER). These can be extracted by computing the distribution of scores for imposters and genuine cases depending on threshold values, as shown in Figure 5.3. The FAR refers to the rate of acceptance imposters as legal persons. The FRR represents the rate of rejection of legal users when the system does not find acceptable matches with current and stored biometric data. The GAR is the rate of the overall accuracy of the system, while the EER reflects the probability of the FAR and FRR scores being equal. It is a measurement of system accuracy, and at a given threshold the better system has the lower ERR score. The GAR, FAR, and FRR are defined as follows:

$$GAR = 100 - FRR. \tag{5.14}$$

$$FAR = \frac{N_{AI}}{N_I} \times 100, \tag{5.15}$$

$$FRR = \frac{N_{RG}}{N_G} \times 100, \tag{5.16}$$

where  $N_{AI}$  is the total number of accepted imposters,  $N_I$  is the total number of imposter claims,  $N_{RG}$  is the total number of rejected genuine claims, and  $N_G$  is the total number of genuine claims.

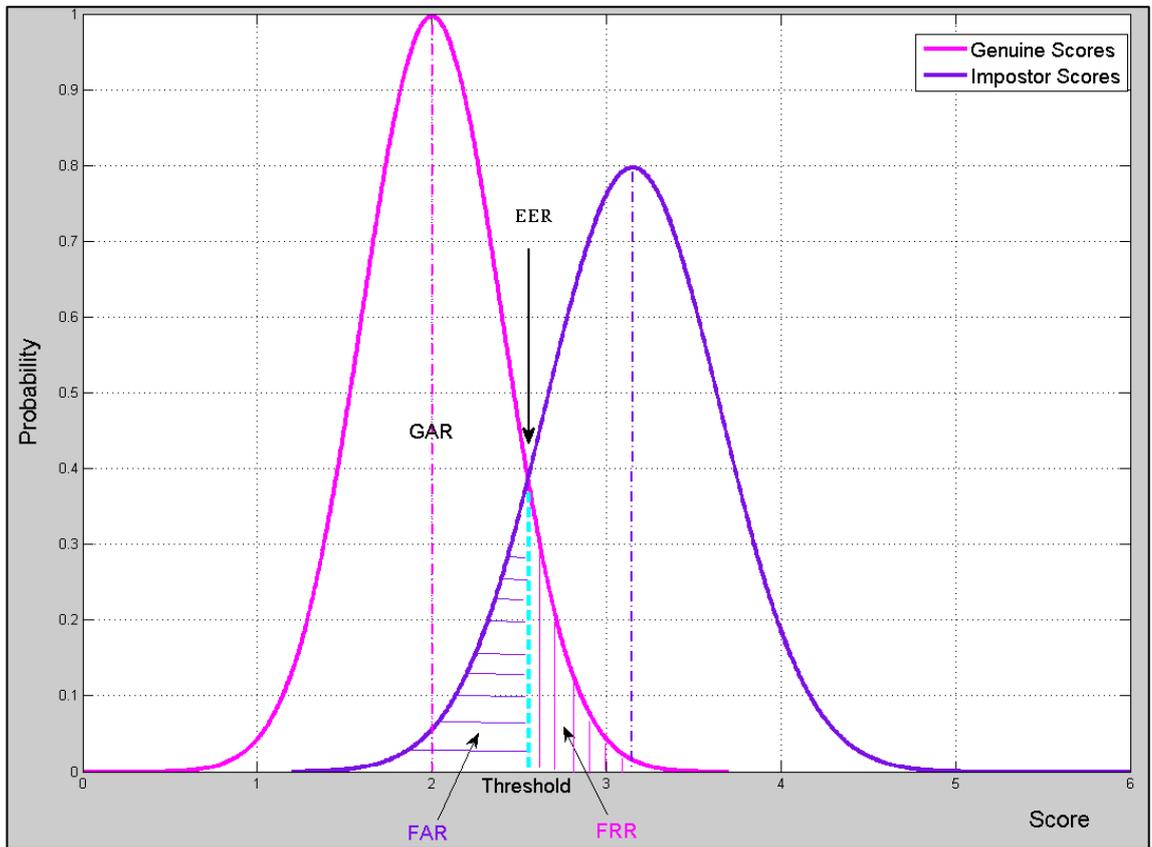


Figure 5.3: Illustration of FRR, FAR, GAR and EER parameters at a given threshold

In the mode of biometric identification, each image in each class is matched with all other images from the same and different classes in order to compute impostor and genuine scores. The matching score is genuine if the images are from the same subject

(class); otherwise, the matching score is an impostor. The final decision is made depending on a threshold value, and decision on incorrect acceptance and rejection is determined as a false acceptance (FA) or false rejection (FR).

In a perfect biometrics system, the maximum threshold value of a genuine score is less than the minimum threshold value of an impostor score. This means that there are no false acceptances or false rejections, and hence there will never be any overlapping region between the scores of impostor and genuine cases. In practice, no biometric system is 100% accurate. Therefore, a more accurate biometric system is one that has a smaller region of overlap. Actually, the overlap is affected by the values of FAR and FRR, which in turn depend on the choice of the classification threshold used. Usually, the setting of the threshold value is significant in making the final decision as to whether a person is rejected as an impostor or accepted as a client. The assigning of the threshold value depends on the usage of a biometrics system at different security levels. In a high-level security biometric system, the threshold should be very high because the desired FAR is very low. On the other hand, the threshold should be low in a low-level security system, because low FRR is desirable. It is clear from Figure 5.3 that if the FRR increases, the FAR decreases, and vice versa, so that the FRR, FAR and GAR are linked to one another.

The minimum FAR and minimum FRR intersect at the EER point by the selection of a certain threshold value. The region under the impostor score but above the threshold point gives the FRR, while the area below the threshold point and under the genuine score gives the FAR. The overall performance of any biometric system should be evaluated in terms of the receiver operating characteristic (ROC). This measures the effect of FAR and FRR, and maps the FAR against the GAR in a graph. Usually, it is used to measure the accuracy of any biometrics matcher at a given level. Therefore, it is used in this chapter to evaluate the accuracy of the proposed MNPP method. The experimental studies in this chapter focus on evaluating both identification and verification modes.

## **5.6 Experiments and Analysis**

This section investigates the properties of the MNPP algorithm. It presents comprehensive set of comparisons with competing algorithms together with the other proposed algorithms in order to evaluate the performance of all of proposed methods. The proposed MNPP method is compared with other popular 1-D methods such as PCA

[15], LDA [14], LPP [31] and NPP [33]. The MNMP is also compared with two-dimensional forms such as 2DPCA [47], 2DLDA [48], 2DLPP [49] and 2DNPP [50]. In addition, it is also compared with the multilinear methods such as MPCA [46], DATER [26], and ND-TSNE [44]. Details of the experiments, and the results and discussions of the MNPP during the identification and verification modes are presented in Sections 5.6.2 and 5.6.3 respectively. The effects of varying dimensions of the reduced space for  $N$ -D methods are investigated in Section 5.4. Finally, the results are discussed in Section 5.5.

### **5.6.1 Databases and experimental design**

In this research, three benchmark facial databases, the ORL [117], AR [118] and FERET [119, 120], are used to evaluate the effectiveness of the proposed MNPP method with unsupervised and supervised learning for the two modes of biometric systems of identification and verification. Details of the three facial databases and the pre-processing method used have been described in Sections 3.6 and 3.7. Experimental studies on identification and verification were performed on all three databases. While different experiments have been used to test the different methods, these experiments are carried out to show the efficiency and effectiveness of the proposed method. All of the experiments were software undertaken using a PENTIUM 4 PC with 3.2 GHz CPU and 4 GB memory and MATLAB [169]. In the experiments, all of training, test and outer tensorial data were transformed into lower-dimensional tensors via the learned subspaces, where the nearest neighbor classifier was used for final classification. The dimensions of each image sample varied depending on which database was used. The ORL database contains only 2-D data, while the AR and FERET databases both contained 3-D images. All facial images from the three databases were manually cropped and normalized to ensure that all tensors were of the same size. For the 1-D methods (PCA, LDA, LPP, and NPP), the images were vectorized as  $4096 \times 1$  vectors. The images were applied directly as  $64 \times 64$  grey scale matrices for the 2-D methods (2DPCA, 2DLDA, 2DLPP and 2DNPP), while they were applied as  $64 \times 64 \times 3$  3-D tensors in the multilinear methods (MPCA, DATER, ND-TSNE and MNPP). However, in the ORL database each image is provided as a 2-D greyscale image, and so  $n=2$  was used for the  $N$ -D methods as a special case in MPCA, DATER, ND-TSNE and MNPP. Thus, for  $N$ -D methods, the images were transformed into the lower space  $p_1 \times p_2$ , where  $p$ ,  $p_1$  and  $p_2$  are the numbers of selected eigenvectors that were tested when  $n=1$  by

varying  $p$  between 1 to 200, and varying both  $p_1$  and  $p_2$  between 2 to 30 when  $n=2$ . For the AR and FERET datasets, when  $n=3$ ,  $p_1 \times p_2 \times p_3$  was used, varying  $p_3$  between 1 and 3.

Different tests were performed on each database using a certain number training samples per subject in each different experiment. The remaining images per subject were used for the testing phase. The training and testing sets were used in both verification and identification modes in order to measure genuine matching scores. In the verification mode, an additional set called the outer set was used to measure the impostor score (FAR), which consisted of testing facial images belonging to the same dataset in addition to facial images from other databases. This explained in more detail in the following sections. Therefore, there were three subsets for each database: training, testing and outer. The training and testing sets were used to evaluate the performance of the MNPP in the identification mode. In the biometrics verification mode, all three sets (training, testing and outer) were used. For  $N_S$  facial images per subject, and  $N_{T_i}$  training images for each subject, there are  $N_W$  various ways to select these  $N_{T_i}$ . The  $N_W$  can be calculated as follows:

$$N_W = \frac{N_S!}{(N_S - N_{T_i})! \times N_{T_i}!} \quad (5.17)$$

Therefore in this chapter, each experiment was repeated 100 times, and the average result computed. The same scenarios were used for the 2DNPP [50], using the same parameters for all of the above methods in order to accurately compare the results of different methods. All the graph-based methods including NPP, LPP, 2DNPP, and 2DLPP are based supervised setting. The LPP and 2DLPP methods use Gaussian weights, therefore the value of the width  $\tau$  of the Gaussian envelop was determined as suggested in [50] as follows: First 1,000 points randomly, and then calculate the pairwise distances among them. It is worth noting that the width  $\tau$  of the Gaussian envelop is set to half the median of those pairwise distances Three main comparison scenarios were used to evaluate the performance of the MNPP in both identification and verifications modes, as follows.



### 5.6.1.2 Experiment on AR Database

In the AR experiments, 100 individual subjects in total were selected, 50 male and 50 female. The first session for each subject was used consisting of 13 samples, in order to test the performance of the proposed MNPP method against other methods. All samples were cropped and normalized to  $64 \times 64 \times 3$  pixels. Sections 5.6.1.2.1 through 5.6.1.2.3 explain the experiments conducted on three probe sets based on different facial expressions, lighting conditions and occlusions, following the same settings as in a previous study [50]. After all these tests, the best performance was selected based on the number of eigenvectors that corresponded to the overall average error rate for all 1-D, 2-D and  $N$ -D methods.

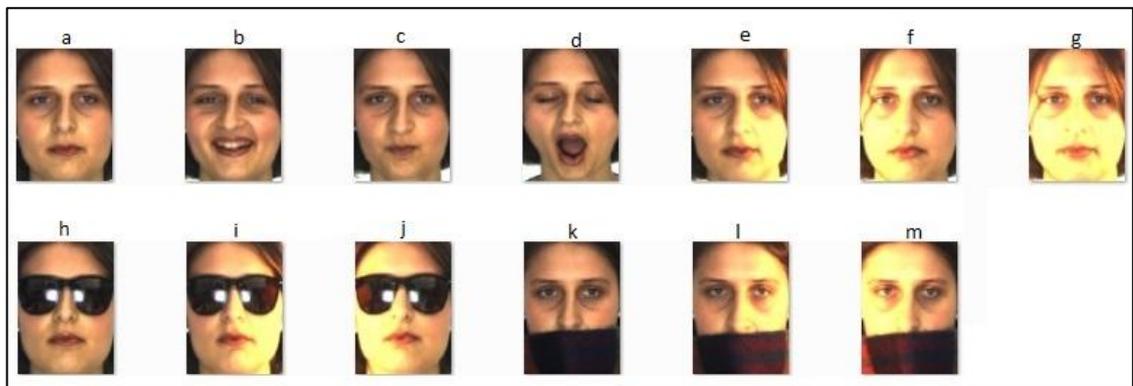


Figure 5.6: AR colour dataset sample with variations in facial expressions, lighting conditions and occlusions.

#### 5.6.1.2.1 AR Database with varying facial expressions (AR1)

Four samples were used in this experiment as shown in Figures 5.6 (a)-(d), from all of the available subjects. Two images were randomly selected from this subset for training with the other two images used for testing. The outer set consists of 2000 images ( $100 \times 99$  classes  $\times$  2 images per class, in addition to 2 FERET images), as shown in Figure 5.7.

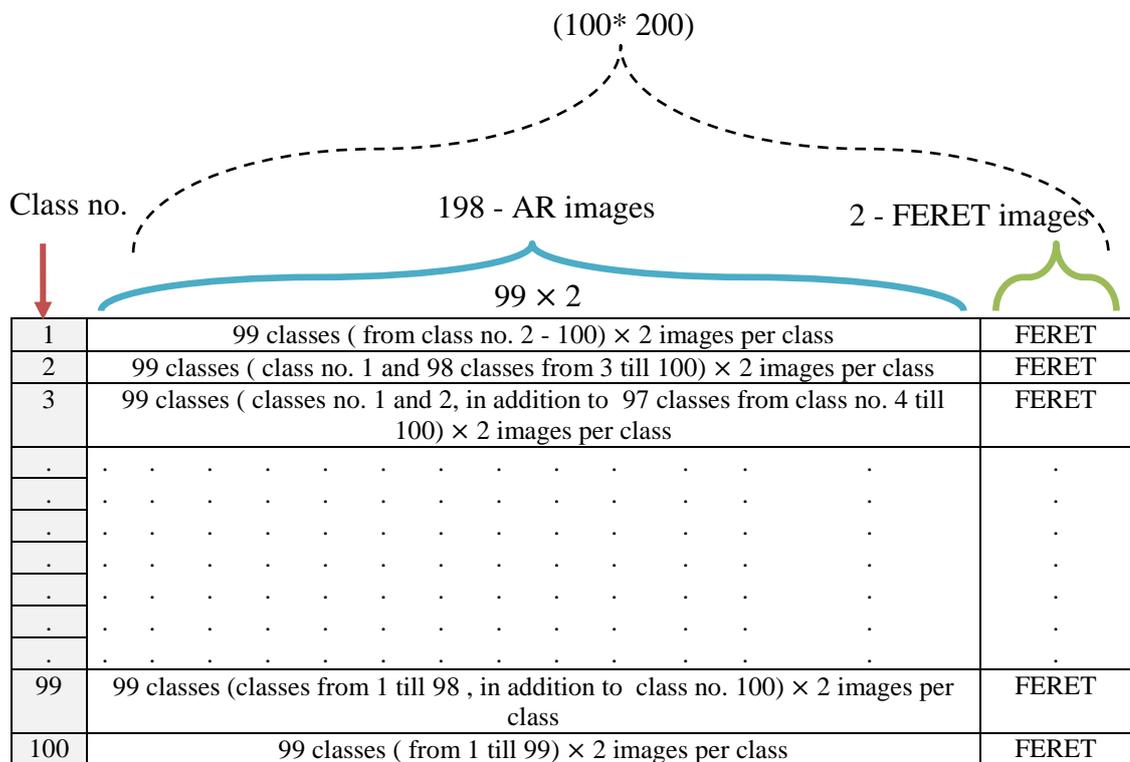


Figure 5.7: Contents of the AR outer set with varying facial expressions

**5.6.1.2.2 AR database with varying lighting conditions (AR2)**

Experiments were also conducted using the AR database with different lighting conditions. The training database samples shown in Figures 5.6 (a), (h) and (k) for each subject were used for training in the different methods, providing a total number of 300 training samples. The images shown in Figures 5.6 (e)-(g), (i), (j), (l), and (m) for each subject were used as the test set, resulting in a total number of images used within this set of 700 images. The outer set consists of 7000 images (100 × 99 classes × 7 images per class, as well as another 7 images from FERET dataset), as shown in Figure 5.8.

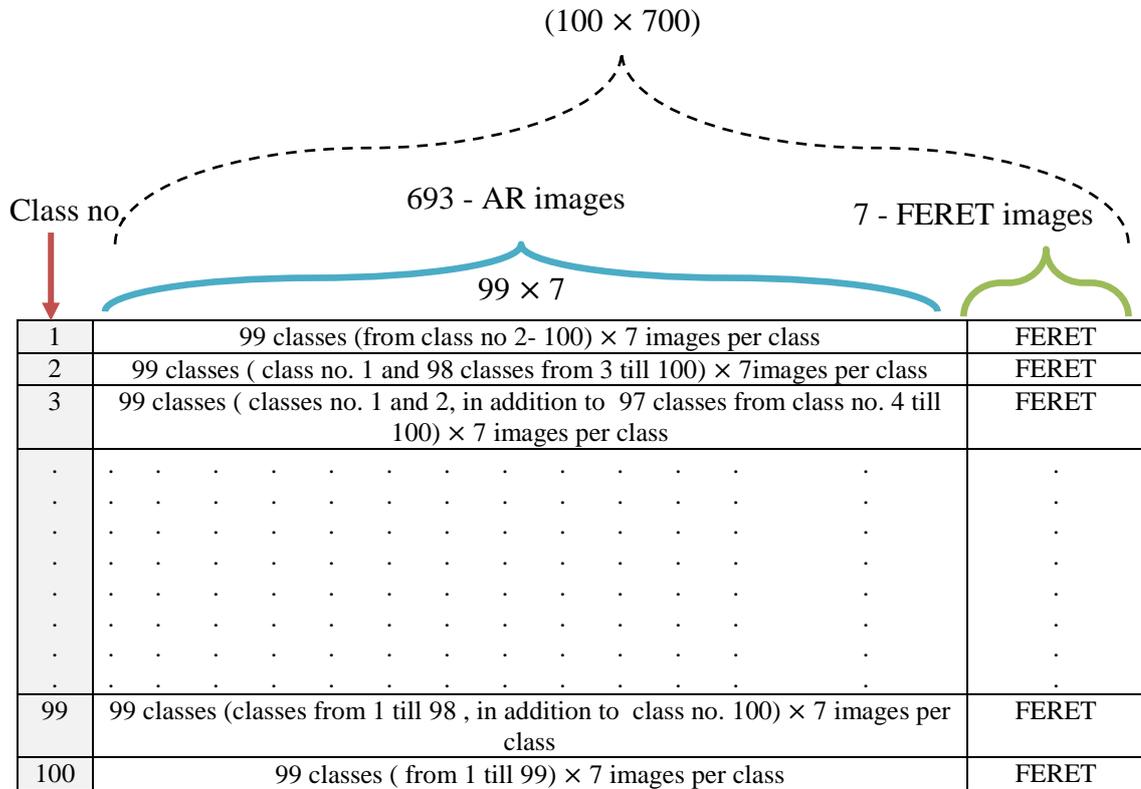


Figure 5.8: Contents of the AR outer set with different lighting conditions

**5.6.1.2.3 AR database under varying occlusions (AR3)**

In the final AR experiment tested the ability of the proposed method to recognize partially occluded faces from the occlusion subset of the AR database. The training set consisted of 600 samples as shown in Figures 5.6 (a)-(f) for each subject. The test images were taken from the samples shown in Figures 5.6 (h)-(m), which in total provided 600 images, while the outer set consisted of 6000 images (100 × 99 classes × 6 images per class, and also 6 images from the FERET dataset), as shown in Figure 5.9.



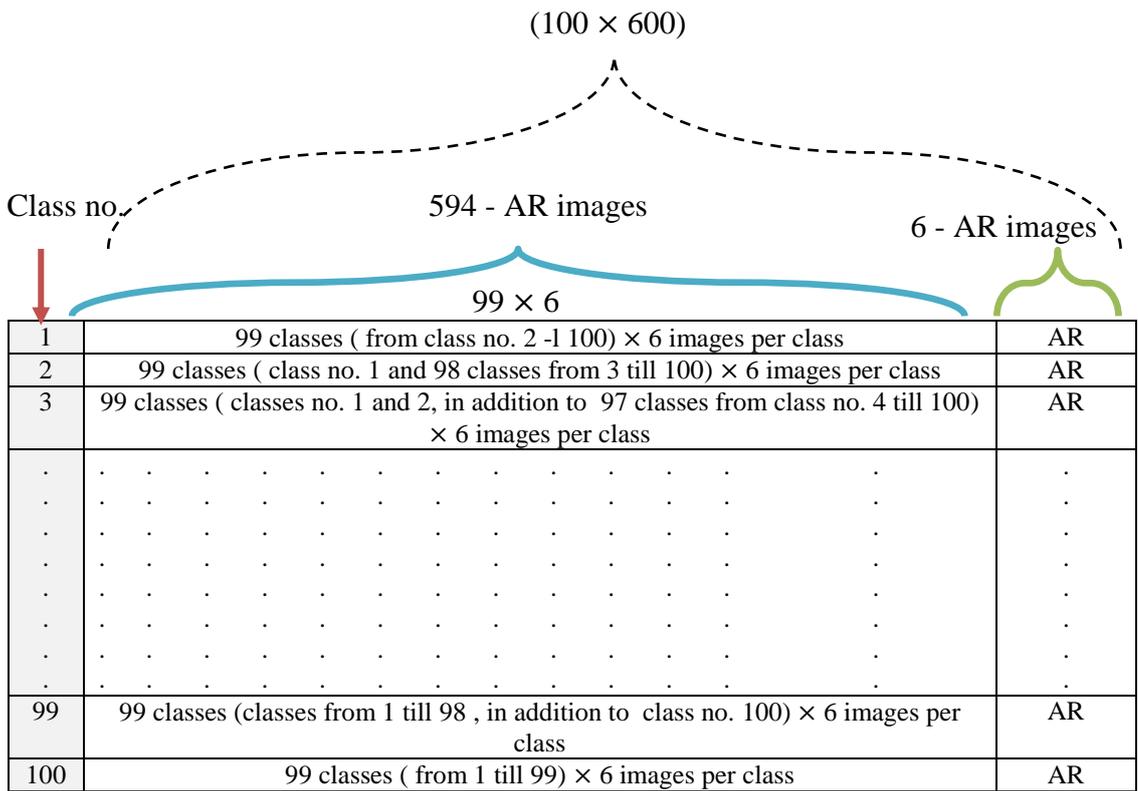


Figure 5.10: Contents of the FERET outer set



Figure 5.11: Facial samples of a subject from the FERET subset

### 5.6.2 Experiments on Identification system

This section presents the results of experiments which represent a comprehensive comparison of MNPP with some competing algorithms (as mentioned above) using the ORL, AR, and FERET. Table 5.2 shows the best error rates from all of the methods used in the experiments on the ORL database. It is clear that the unsupervised and supervised MNPP achieves lowest error rate of all other methods, including the 2DNPP, using a reasonable and acceptable number of eigenvectors and giving the fewest wrongly classified images. In terms of error rates the proposed method in unsupervised and supervised modes yields decreases in error rates of more than 16% and 41% compared to DATER respectively, and more than 30% and 50% over 2DNPP.

Table 5.2: Performance comparisons of the error rate (Mean  $\pm$  SD%) using the ORL database

Method	$p^{(n=1)}$	$p_1^{n=2}$	$p_2^{n=2}$	Error rate (%)
PCA	41	-	-	15.5 $\pm$ 2.6
LDA	30	-	-	8.8 $\pm$ 2.1
LPP	54	-	-	14.2 $\pm$ 2.7
NPP	150	-	-	13.7 $\pm$ 1.9
2DPCA	-	5	64	10.5 $\pm$ 2.5
2DLDA	-	5	64	6.4 $\pm$ 1.5
2DLPP	-	5	64	6.8 $\pm$ 2.6
<i>2DNPP</i>	-	9	64	5.9 $\pm$ 1.4
<b>C-B2DNPP</b>	-	<b>6</b>	<b>4</b>	<b>3.5 <math>\pm</math> 0.5</b>
MPCA	-	15	20	8.5 $\pm$ 2.3
DATER	-	14	10	4.9 $\pm$ 1.2
ND-TSNE	-	11	12	5.4 $\pm$ 0.7
<b>MNPP</b>	-	<b>13</b>	<b>13</b>	<b>4.1 <math>\pm</math> 0.5</b>
<b>MNPP (Supervised)</b>	-	<b>13</b>	<b>13</b>	<b>2.9 <math>\pm</math> 0.3</b>

Table 5.3 Tables 5.3 and 5.4 illustrate the comparative results using the AR dataset with varying facial expressions, showing that the unsupervised MNPP obtained better results than most of the other methods, including the 2DNPP, although the 2DLDA and DATER performed slightly better. However, the supervised MNPP results are superior to all of the other methods. Here the performance gains of the supervised approach represent substantial decreases in error rate compared to DATER, where gray level images and colour images the savings are 32% and 44% respectively.

Table 5.3: Performance comparisons of the error rate (Mean  $\pm$  SD%) using the AR database under varying facial expressions

Method	$p^{(n=1)}$	$p_1^{n=2}$	$p_2^{n=2}$	Error rate (%)
PCA	110	-	-	13.5 $\pm$ 2.8
LDA	50	-	-	9.9 $\pm$ 2.3
LPP	30	-	-	19.9 $\pm$ 2.5
NPP	30	-	-	15 $\pm$ 2.7
2DPCA	-	10	64	8.2 $\pm$ 2.1
2DLDA	-	6	64	2.2 $\pm$ 2.6
2DLPP	-	5	64	6.8 $\pm$ 2.3
2DNPP	-	5	64	3.7 $\pm$ 2.2
<b>C-B2DNPP</b>	-	<b>9</b>	<b>3</b>	<b>2.0 <math>\pm</math> 0.5</b>
MPCA	-	15	25	6.5 $\pm$ 2.1
DATER	-	10	20	1.8 $\pm$ 0.5
ND-TSNE	-	12	15	4.6 $\pm$ 1.5
<b>MNPP</b>	-	<b>9</b>	<b>20</b>	<b>2.2 <math>\pm</math> 0.6</b>
<b>MNPP (Supervised)</b>	-	<b>9</b>	<b>20</b>	<b>1.5 <math>\pm</math> 0.4</b>

Table 5.4: Performance comparisons of the error rate (Mean  $\pm$  SD%) using the AR colour database under varying facial expressions

Method	$p_1^{n=3}$	$p_2^{n=3}$	$p_3^{n=3}$	Error rate (%)
MPCA	15	25	2	$5.4 \pm 1.5$
DATER	10	20	2	$1.3 \pm 0.2$
ND-TSNE	12	15	2	$2.9 \pm 1.1$
<b>MNPP</b>	<b>9</b>	<b>20</b>	<b>2</b>	<b><math>1.6 \pm 0.5</math></b>
<b>MNPP (Supervised)</b>	<b>9</b>	<b>20</b>	<b>2</b>	<b><math>0.8 \pm 0.2</math></b>

Tables 5.5 and 5.6 reported the experimental results on AR database under varying lighting conditions, showing that the supervised and unsupervised MNPP out-performs all of the other methods even 2-D and *N*-D extensions of the LDA (2DLDA and DATER), achieving over 90% better than DATER in these cases, and approximately 76 and 67% improvement in comparison with the ND-TSNE and 2DNPP respectively. The 1-D, 2-D and *N*-D performance of the PCA is the worst method compared with all other methods used within this experiment.

The detection of occlusions in facial images appears to be an extremely difficult task. However, Tables 5.7 and 5.8 show that both the supervised and unsupervised MNPP with the AR database images including varying occlusions still perform better than all of the other methods. The improvement in recognition performance is over 10% compared to the 2DNPP, DATER and ND-TSNE for the unsupervised MNPP, and more than 37% for the supervised MNPP, while the MPCA appeared to be the worst method in this experiment. The reason for the lack of performance of the MPCA will be discussed later in section 5.5.

Table 5.5: Performance comparisons of the error rate (Mean  $\pm$  SD%) using the AR database with varying lighting conditions

Method	$p^{(n=1)}$	$p_1^{n=2}$	$p_2^{n=2}$	Error rate (%)
PCA	180	-	-	77.2 $\pm$ 2.9
LDA	90	-	-	8.6 $\pm$ 1.5
LPP	80	-	-	21.2 $\pm$ 1.9
NPP	130	-	-	16.4 $\pm$ 2.5
2DPCA	-	25	64	49.4 $\pm$ 2.7
2DLDA	-	25	64	27.4 $\pm$ 2.6
2DLPP	-	9	64	11.2 $\pm$ 2.1
2DNPP	-	12	64	4.9 $\pm$ 1.8
<b>C-B2DNPP</b>	-	<b>6</b>	<b>4</b>	<b>3.5 <math>\pm</math> 0.5</b>
MPCA	-	25	25	45.8 $\pm$ 2.7
DATER	-	10	15	21.5 $\pm$ 2.9
ND-TSNE	-	10	25	8.8 $\pm$ 1.3
<b>MNPP</b>	-	<b>8</b>	<b>20</b>	<b>2.1 <math>\pm</math> 0.9</b>
<b>MNPP (Supervised)</b>		<b>8</b>	<b>20</b>	<b>1.6 <math>\pm</math> 0.5</b>

Table 5.6: Performance comparisons of the error rate (Mean  $\pm$  SD%) using the AR colour database with varying lighting conditions

Method	$p_1^{n=3}$	$p_2^{n=3}$	$p_3^{n=3}$	Error rate (%)
MPCA	25	25	2	43.2 $\pm$ 2.6
DATER	10	15	2	20.1 $\pm$ 2.3
ND-TSNE	10	25	2	5.9 $\pm$ 1.2
<b>MNPP</b>	<b>8</b>	<b>20</b>	<b>2</b>	<b>0.9 <math>\pm</math> 0.5</b>
<b>MNPP (Supervised)</b>	<b>8</b>	<b>20</b>	<b>2</b>	<b>0.6 <math>\pm</math> 0.3</b>

Table 5.7: Performance comparisons of the error rate (Mean  $\pm$  SD%) using the AR database with varying occlusion

Method	$p^{(n=1)}$	$p_1^{n=2}$	$p_2^{n=2}$	Error rate (%)
PCA	180	-	-	75.3 $\pm$ 2.8
LDA	100	-	-	51.1 $\pm$ 2.7
LPP	150	-	-	91.4 $\pm$ 2.9
NPP	70	-	-	74.7 $\pm$ 2.1
2DPCA	-	20	64	70.3 $\pm$ 2.6
2DLDA	-	15	64	44.7 $\pm$ 2.5
2DLPP	-	13	64	46.9 $\pm$ 2.4
2DNPP	-	15	64	42.1 $\pm$ 2.7
<b>C-B2DNPP</b>	-	<b>9</b>	<b>6</b>	<b>37.5 <math>\pm</math> 1.5</b>
MPCA	-	15	25	68.7 $\pm$ 2.8
DATER	-	10	20	41.9 $\pm$ 1.1
ND-TSNE	-	17	25	43.5 $\pm$ 1.5
<b>MNPP</b>	-	<b>11</b>	<b>25</b>	<b>39.2 <math>\pm</math> 1.3</b>
<b>MNPP (Supervised)</b>		<b>11</b>	<b>25</b>	<b>27.4 <math>\pm</math> 1.2</b>

Table 5.8: Performance comparisons of the error rate (Mean  $\pm$  SD%) using the AR colour database with varying occlusion

Method	$p_1^{n=3}$	$p_2^{n=3}$	$p_3^{n=3}$	Error rate (%)
MPCA	15	25	2	66.1 $\pm$ 2.8
DATER	10	20	2	38.5 $\pm$ 1.2
ND-TSNE	17	25	2	39.9 $\pm$ 2.1
<b>MNPP</b>	<b>11</b>	<b>25</b>	<b>2</b>	<b>35.7 <math>\pm</math> 1.3</b>
<b>MNPP (Supervised)</b>	<b>11</b>	<b>25</b>	<b>2</b>	<b>24.9 <math>\pm</math> 1.4</b>

The detailed results given in Tables 5.9 and 5.10 show that the supervised and unsupervised MNPP both give superior performance and smaller error rates than all of the other methods, with the least number of eigenvectors used. Here the performance gains in for the proposed supervised method in terms of decreases in error rate over ND-

TSNE are substantial, where in gray level and colour images the decreases are is 32% and 44% respectively, and more than 37% compared with 2DNPP.

Table 5.9: Performance comparisons of the error rate (Mean  $\pm$  SD%) using the FERET gray database

Method	$p^{(n=1)}$	$p_1^{n=2}$	$p_2^{n=2}$	Error rate (%)
PCA	80	-	-	43.9 $\pm$ 2.7
LDA	50	-	-	30.6 $\pm$ 2.5
LPP	65	-	-	30.8 $\pm$ 2.4
NPP	90	-	-	28.5 $\pm$ 2.8
2DPCA	-	13	64	41.8 $\pm$ 2.5
2DLDA	-	3	64	26.4 $\pm$ 2.3
2DLPP	-	5	64	26.5 $\pm$ 2.0
2DNPP	-	9	64	19.7 $\pm$ 2.2
<b>C-B2DNPP</b>	-	<b>6</b>	<b>4</b>	<b>14.2 <math>\pm</math> 0.5</b>
MPCA	-	15	25	30.1 $\pm$ 2.7
DATER	-	12	15	19.3 $\pm$ 2.1
ND-TSNE	-	11	16	172 $\pm$ 0.5
<b>MNPP</b>	-	<b>14</b>	<b>4</b>	<b>16.0 <math>\pm</math> 0.5</b>
<b>MNPP (Supervised)</b>		<b>14</b>	<b>4</b>	<b>12.3 <math>\pm</math> 0.3</b>

Table 5.10: Performance comparisons of the error rate (Mean  $\pm$  SD%) using the FERET colour database

Method	$p_1^{n=3}$	$p_2^{n=3}$	$p_3^{n=3}$	Error rate (%)
MPCA	15	25	2	28.4 $\pm$ 2.2
DATER	12	15	2	16.6 $\pm$ 1.8
ND-TSNE	11	16	2	15.8 $\pm$ 1.1
<b>MNPP</b>	<b>14</b>	<b>4</b>	<b>2</b>	<b>14.1 <math>\pm</math> 0.5</b>
<b>MNPP (Supervised)</b>	<b>14</b>	<b>4</b>	<b>2</b>	<b>10.9 <math>\pm</math> 0.3</b>

### 5.6.3 Experiments on the verification system

The verification tests determined the ability of the MNPP and other algorithms to correctly accept or reject the claimed identities of users. The performance of systems was measured according to how many times decisions to accept or reject were made correctly. An acceptance or rejection decision may depend on the threshold value chosen, and MNPP calculated the minimum distance between the training and testing images using the nearest neighbor classifier. Thus, in this research, the threshold values determined depending of the features extracted from MNPP and some of other existing methods on the three databases ORL, AR, and FERET can be defined as follows:

1. Find the minimum and maximum distances for all normal faces in the testing dataset this is called the testing distance.
2. Find the minimum and maximum distances for all stranger faces in the outer dataset, which is called the outer distance.
3. Set a range of acceptance threshold values as the minimum and maximum values of the testing distance.
4. Set a range of rejection threshold values as the minimum and maximum values of the outer distance.

In practice, it is very hard in any biometric system to find features which can be used to 100% accurately discriminate between images from the testing database and the outer database. Therefore the value of the minimum outer distance is usually set at less than the maximum value of the testing distance. So, some legal users may be classified as rejections (FRs), and other illegitimate persons could be classified as acceptances (FAs). Ideally, both FRR and FAR should be zero. But in practice, changing the threshold value can result in reducing either the FRR and FAR and increasing the other.

In these experiments, the threshold values were varied through the minimum testing distance to the maximum outer testing in order to compare the complete operating range of the proposed MNPP method and other existing algorithms. In each method, a pair of FAR and FRR values were produced as the settings of each threshold value, which were plotted on a graph using the ROC. Usually, the ROC curve plots the FAR against the genuine acceptance rate ( $GAR=100-FRR$ ) instead of FRR. As such for each database,

the verification error rates (FAR and FRR) were computed for various threshold values and the ROCs are plotted to compare the results.

The experiments are compared the proposed MNPP algorithm with state of the art methods such as NPP, 2DNPP, MPCA, and DATER using the ORL, AR, and FERET databases. For grey level facial images, the MNPP was compared with the NPP and 2DNPP methods. Tables 5.11-5.15 show EER and GAR values for the lowest FAR using the grey scale facial images of all three databases in the different scenarios explained in Section 5.6.1. Figures 5.12, 5.14, 5.16, 5.18, and 5.20 illustrate the comparisons of values of true acceptance (TA), false rejection (FR), and false acceptance (FA) of the MNPP and the NPP, 2DNPP methods with the three databases. Figures 5.13, 5.15, 5.17, 5.19 and 5.20 display the comparisons of the receiver operating characteristics (ROC) curves for the MNPP method and the NPP and 2DNPP algorithms. At the lowest values of FAR, Table 5.11 shows that the proposed MNPP method improved the equal error rate (EER) by about 34.6%, and 21.9% respectively compared with the NPP and 2DNPP with ORL database. Figure 5.12 shows that the numbers of TA and FR for the MNPP are better than those for both NPP and 2DNPP methods with the ORL database. Figure 5.13 shows that the genuine acceptance rate (GAR) for the MNPP is superior to those of both NPP and 2DNPP for this database. Using MNPP, the GAR is increased by 10.8% and 3% respectively compared with NPP and 2DNPP.

Table 5.11: EER and GAR of the NPP, 2DNPP and the proposed MNPP method with the ORL database

Method	EER	GAR at the lowest FAR (0.1)
NPP	0.98	86.5
2DNPP	0.82	94.0
MNPP	0.64	97.0

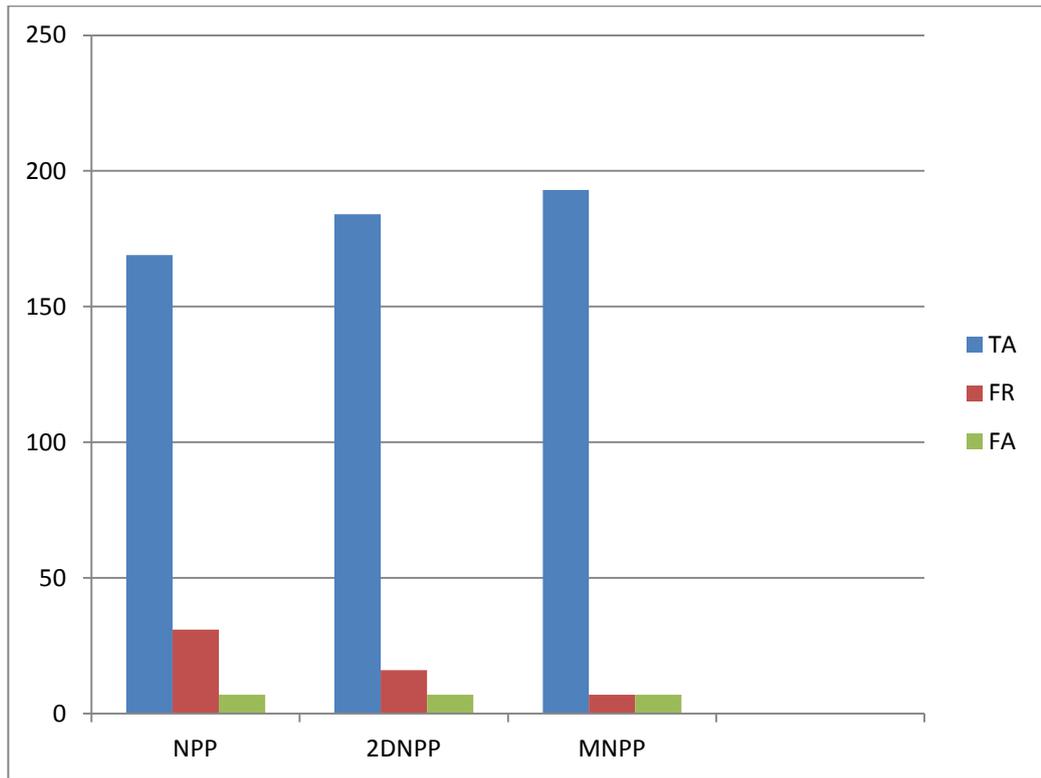


Figure 5.12: Comparisons of values of TA, FR, and FA the of the NPP, 2DNPP and the proposed MNPP method with the ORL database

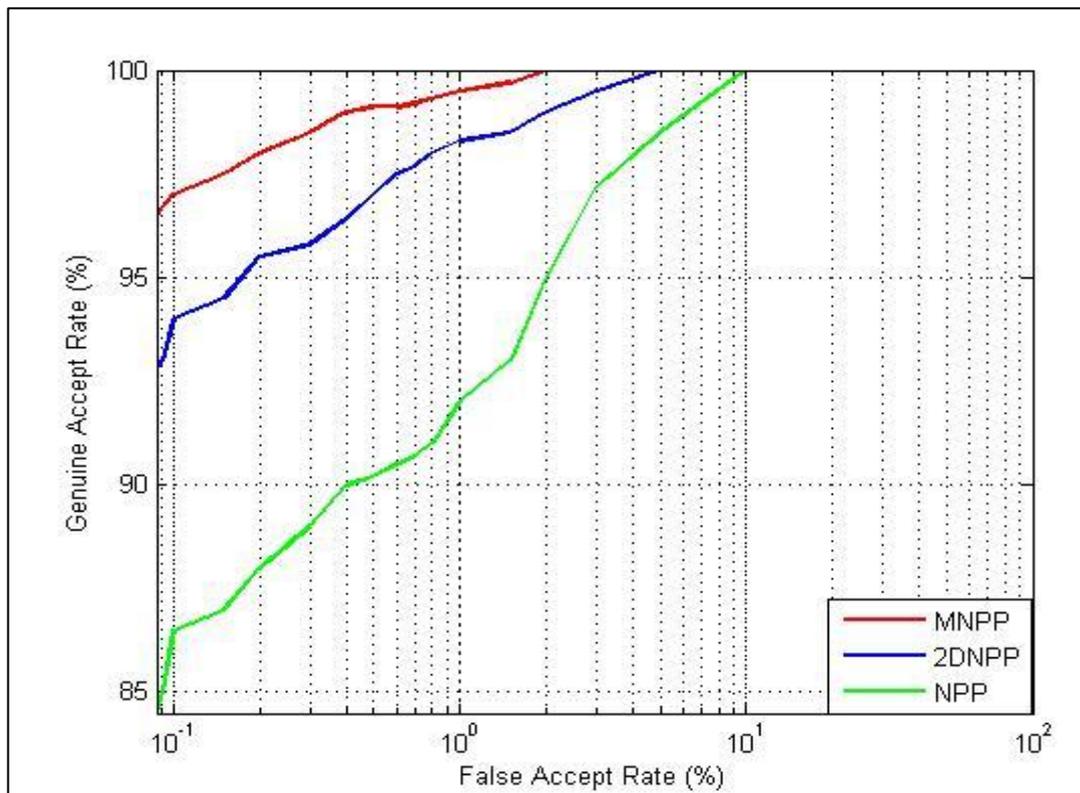


Figure 5.13: Comparisons of the ROC curves for the NPP, 2DNPP and proposed MNPP method with ORL database.

From the experiments with the AR1 database, the EER, of the NPP and 2DNPP methods can be reduced by 56.0% and 12.1% respectively using the MNPP approach, as shown in Table 5.12. The TA and FR values for the MNPP are more than those for NPP and 2DNPP methods, as shown in Figure 5.14. The GAR at the lowest false acceptance rates for the MNPP is improved by about 10.7% and 1% respectively compared to NPP and 2DNPP, as shown in Figure 5.13.

Table 5.12: Values of EER and GAR for the NPP, 2DNPP and the proposed MNPP method with AR grey database and varying facial expressions

Method	EER	GAR at the lowest FAR (0.1)
NPP	0.82	87.9
2DNPP	0.41	97.5
MNPP	0.36	98.5

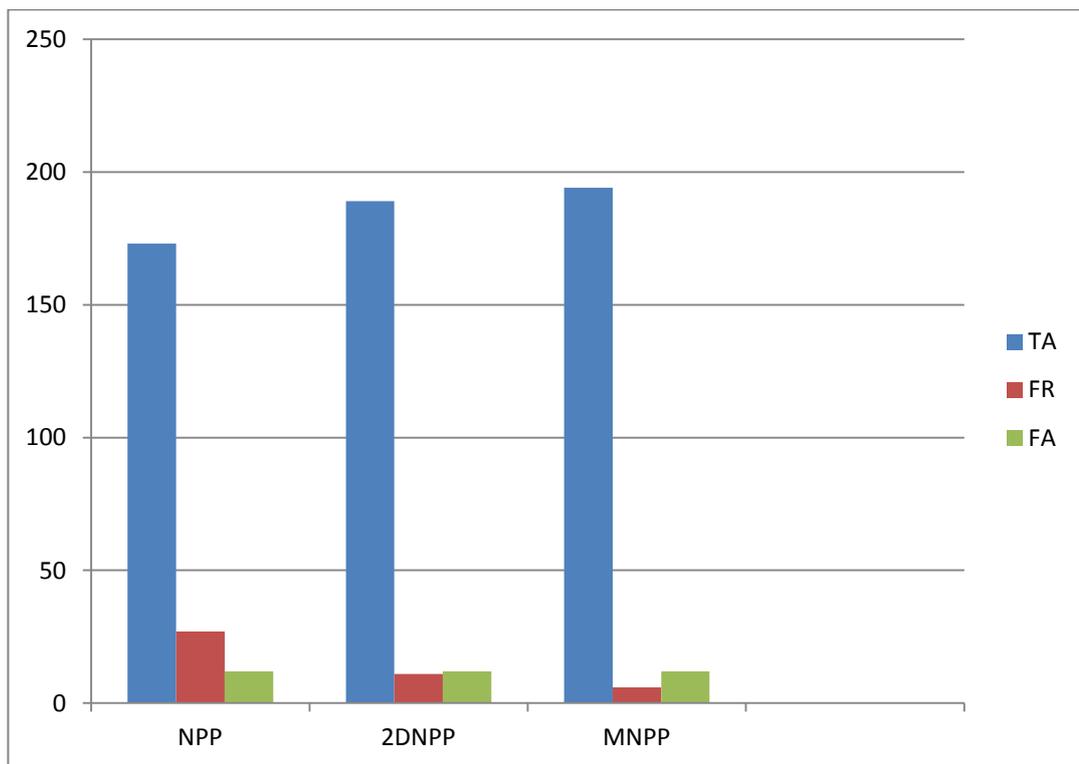


Figure 5.14: Comparisons of TA, FR, and FA values for the NPP, 2DNPP and proposed MNPP method with AR grey database and varying facial expressions

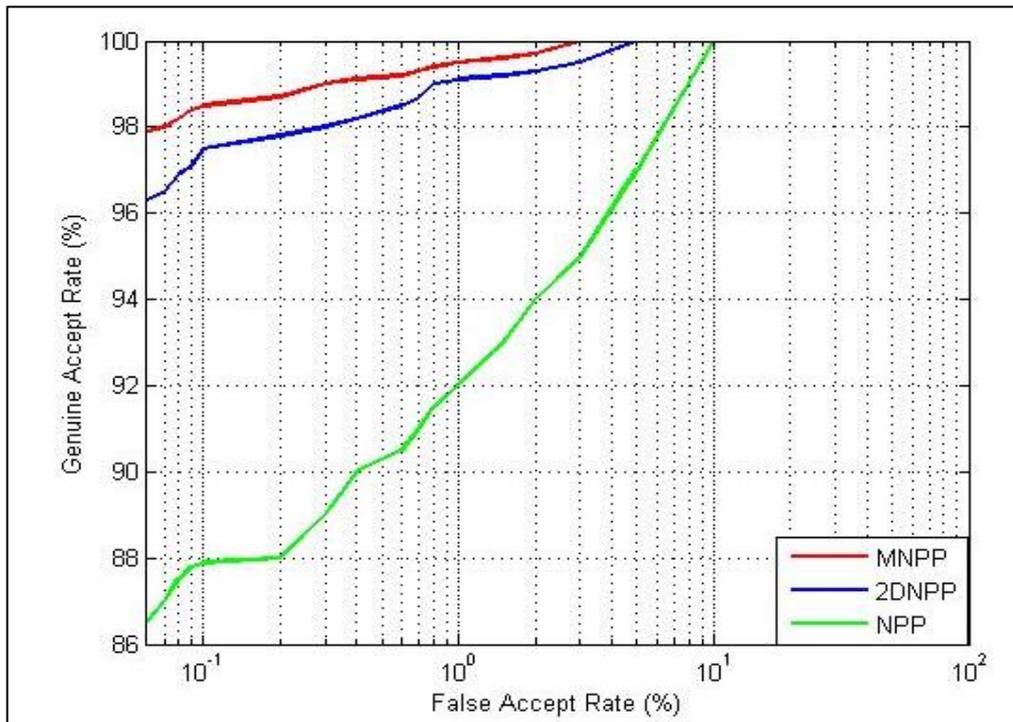


Figure 5.15: Comparisons of ROC curves for the NPP, 2DNPP and proposed MNPP method with the AR grey database and varying facial expressions

From the experiments with the of AR2 database, Table 5.13 shows that the EER values of the NPP and 2DNPP methods can be improved by 82.3% and 39.0% respectively using the proposed MNPP method. Figure 5.16 shows that the TA and FR values for the MNPP are better than those for NPP and 2DNPP methods. Figure 5.17 proves that the GAR at the lowest false acceptance rates for the MNPP is greater than those of both NPP and 2DNPP. The MNPP yields about 12.1% and 3.2% improvements respectively compared to the NPP and 2DNPP.

Table 5.13: Values of EER and GAR for the NPP, 2DNPP and proposed MNPP method with the AR grey database and varying lighting conditions

Method	EER	GAR at the lowest FAR (0.1)
NPP	1.64	86.3
2DNPP	0.93	95.1
MNPP	0.29	98.2

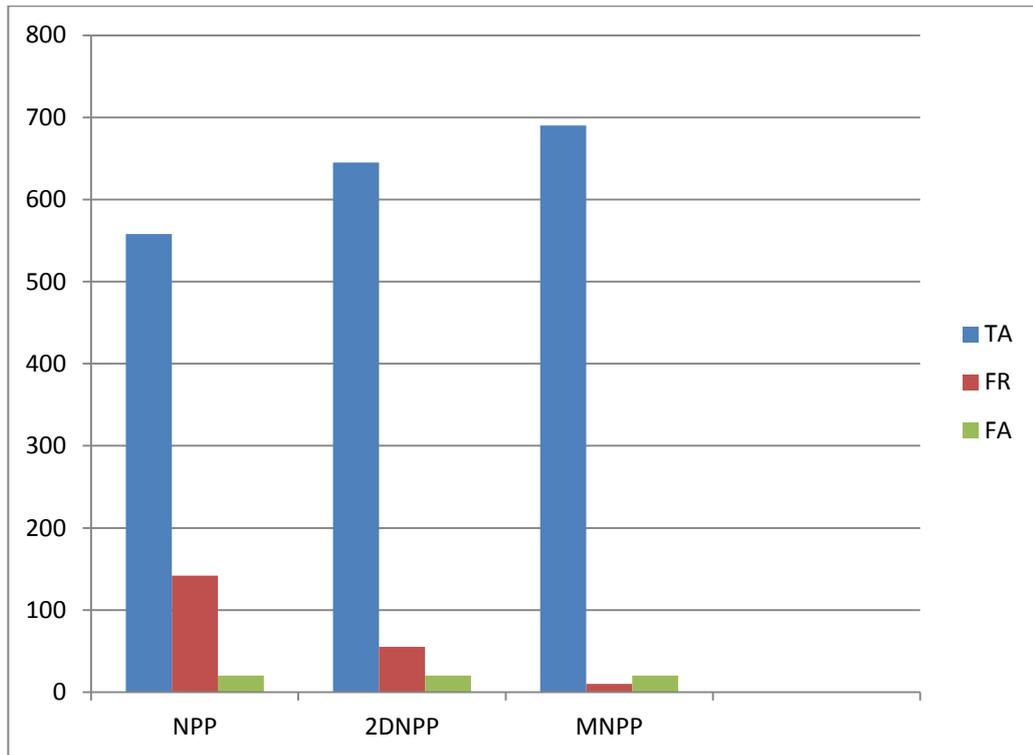


Figure 5.16: Comparisons of TA, FR, and FA values for the NPP, 2DNPP and proposed MNPP method with the AR grey database and varying lighting conditions

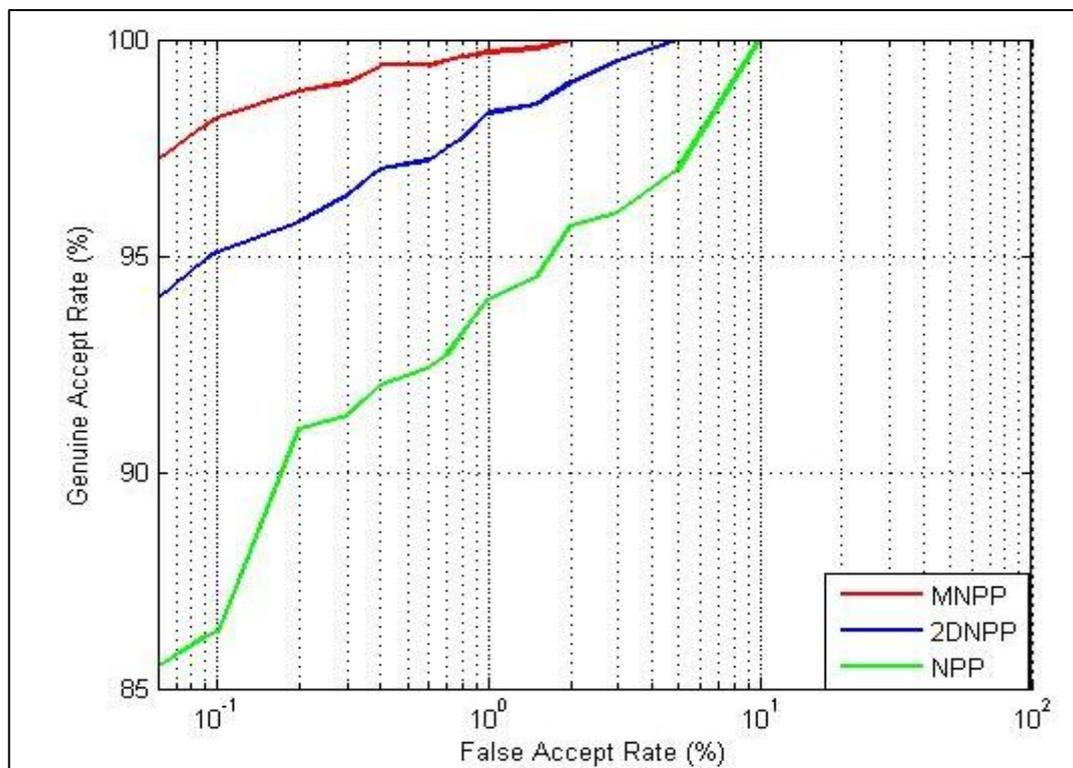


Figure 5.17: Comparisons of the ROC curves for the NPP, 2DNPP and the proposed MNPP method with the AR grey database and varying lighting conditions

The results of the experiments with the of AR3 database at the lowest FAR show that the proposed MNPP method enhances the EER by about 84.5%, and 24.8% respectively compared with the NPP and 2DNPP, as shown in Table 5.14. Figure 5.18 illustrates that the values of TA and FR for the MNPP are much better than those for the NPP and also the 2DNPP methods. The GAR for the MNPP is increased by 72.3% and 26.6% respectively compared with those of both NPP and 2DNPP methods respectively, as shown in Figure 5.19.

Table 5.14: Values of EER and GAR for the NPP, 2DNPP and proposed MNPP method with the AR grey database and varying occlusion

Method	EER	GAR at the lowest FAR (0.5)
NPP	9.00	19.5
2DNPP	1.85	51.7
MNPP	1.39	70.5

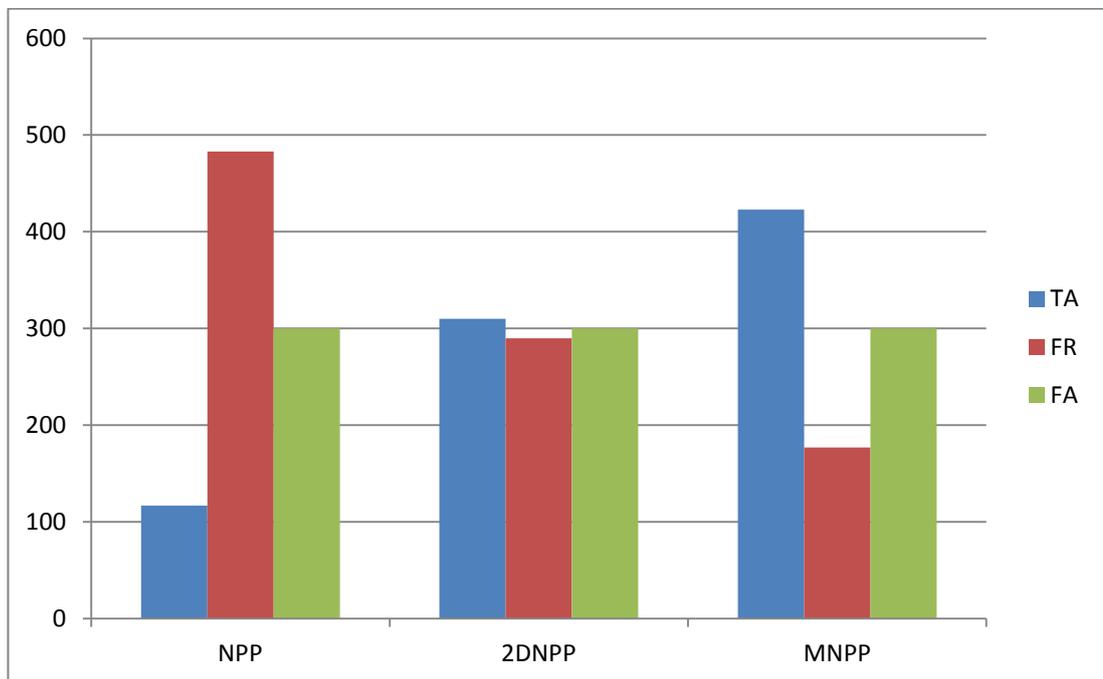


Figure 5.18: Comparisons of TA, FR, and FA values for the NPP, 2DNPP and the proposed MNPP method with the AR grey database and varying occlusion

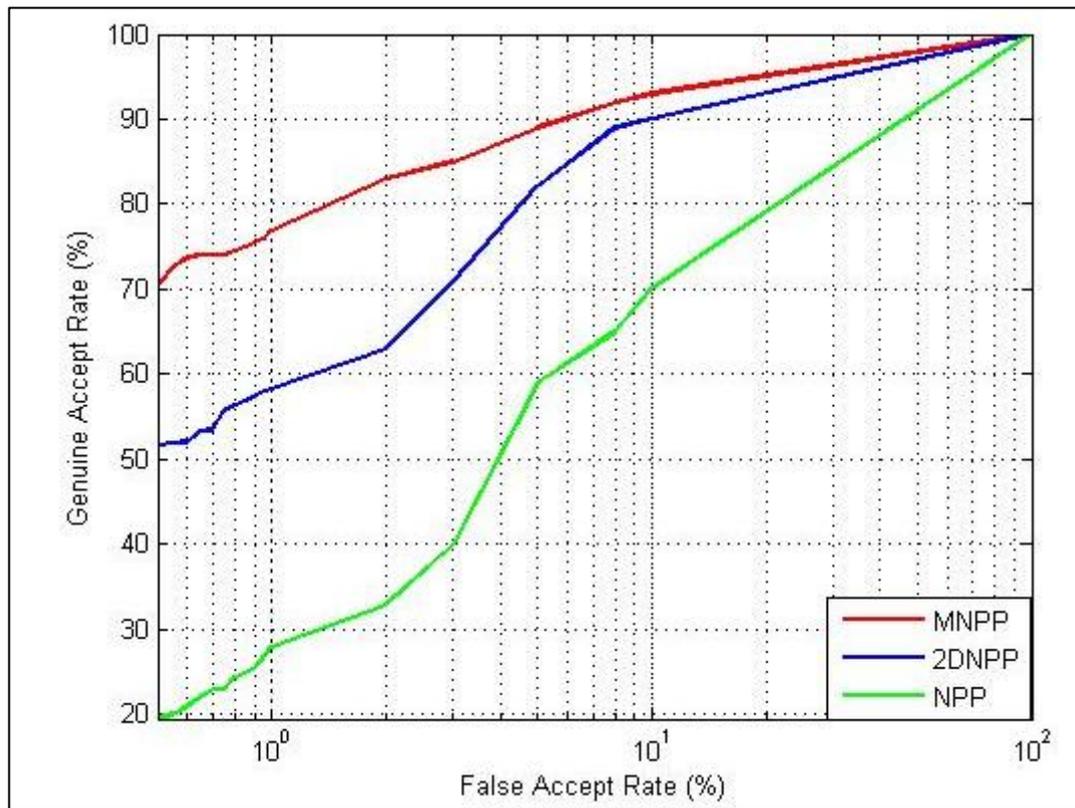


Figure 5.19: Comparisons of ROC curves for the NPP, 2DNPP and the proposed MNPP method with the AR grey database and varying occlusion

The results of experiments with the of FERET database indicate that the EER of the MNPP method is improved by 57.7% and 55.3% respectively more than the EER of NPP and 2DNPP methods respectively, as explained in Table 5.15. Figure 5.20 shows that the TA and FR numbers using the MNPP are greater than those for both NPP and 2DNPP methods. Figure 5.21 proved that the GAR at the lowest false acceptance rates for the MNPP is increased by about 24.7% and 9.1% compared with NPP and 2DNPP algorithms, respectively.

Table 5.15: Values of EER and GAR for the NPP, 2DNPP and the proposed MNPP method with the FERET grey database

Method	EER	GAR at the lowest FAR (0.1)
NPP	2.25	65.5
2DNPP	2.13	79.0
MNPP	0.95	87.0

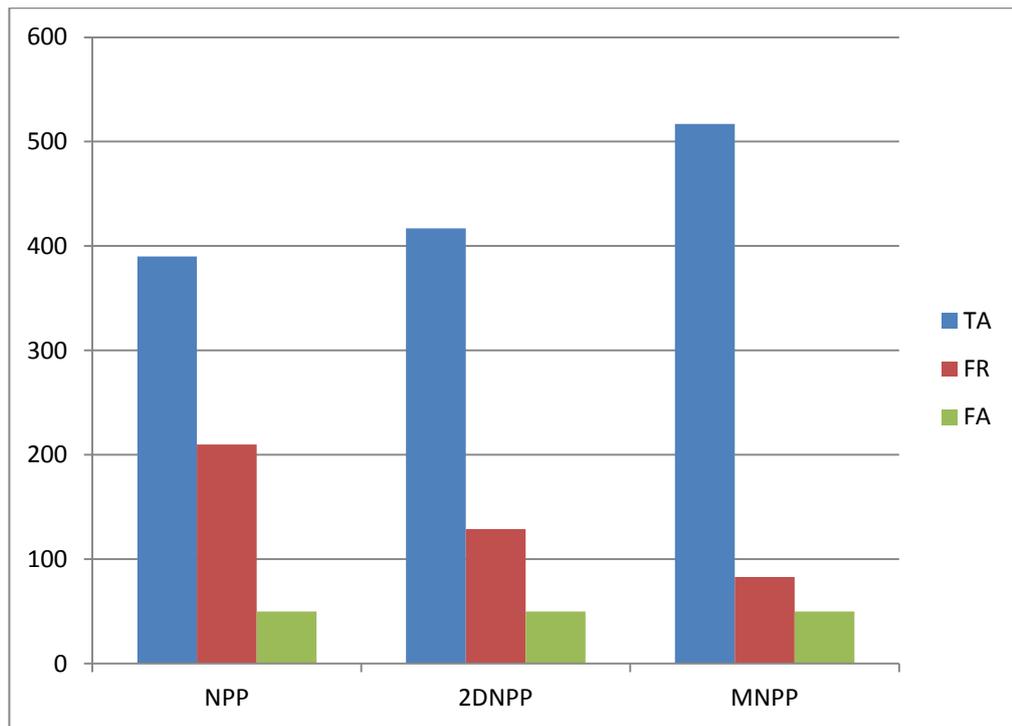


Figure 5.20: Comparisons of TA, FR, and FA values for the NPP, 2DNPP and the proposed MNPP method with the FERET grey database.

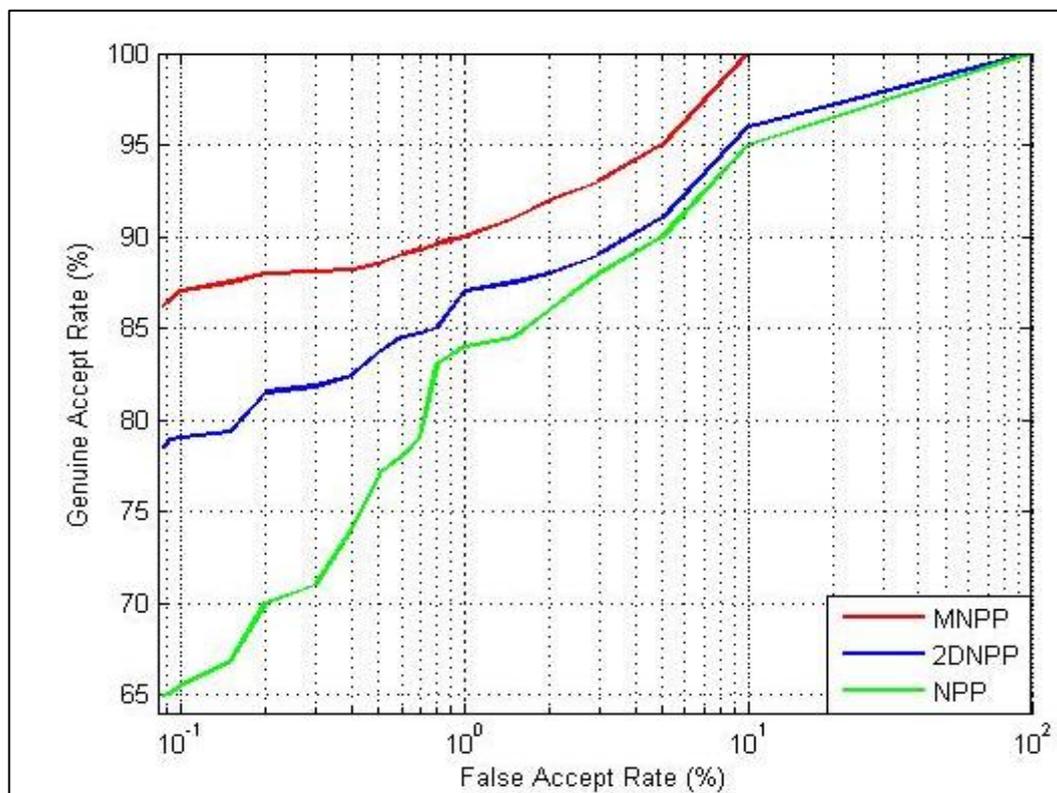


Figure 5.21: Comparisons of ROC curves for the NPP, 2DNPP and the proposed MNPP method with the FERET grey database.

In this chapter, the MNPP is compared with the MPCA and DATER methods in experiments using colour scale facial images. The three rates of EER and GAR at the lowest values of FAR are shown in Tables 5.16 and 5.17 using the colour level facial images from the databases. The comparisons of values of true acceptance (TA), false rejection (FR), and false acceptance (FA) of the MNPP against those of the MPCA, and DATER methods with the three databases are presented in Figures 5.22, 5.24, 5.26, and 5.28. Figures 5.23, 5.25, 5.27 and 5.29 display the ROC curves that obtained using the MPCA, DATER and MNPP. Tables 5.16-5.19 show that the proposed MNPP method gained Equal Error Rates (EER) of 65.1%, 90.8%, 83.1% and 69.2% respectively compared with MPCA with the AR1, AR2, AR3, and FERET datasets. The EER of the DATER decreased by 11.1%, 84.5%, 59.7%, and 48.5% respectively with the AR1, AR2, AR3, and FERET, as presented in Figures 5.22, 5.24, 5.26, and 5.28. The GAR for the MNPP is better than those of both NPP and 2DNPP, as illustrated in Figures 5.23, 5.25, 5.27 and 5.29. The MNPP gained approximately 7.1%, 44.6%, 70.0%, and 17.0% improvement respectively compared with MPCA with the AR1, AR2, AR3, and FERET databases. Whereas, it achieved about 0.5%, 19.7%, 18.0%, and 6.6% improvement respectively compared with DATER using the AR1, AR2, AR3, and FERET databases.

Table 5.16: Values of EER and GAR for the MPCA, DATER and proposed MNPP method on AR colour database and varying facial expressions

Method	EER	GAR at the lowest FAR (0.1)
MPCA	0.86	91.0
MLDA	0.30	97.5
MNPP	0.27	98.0

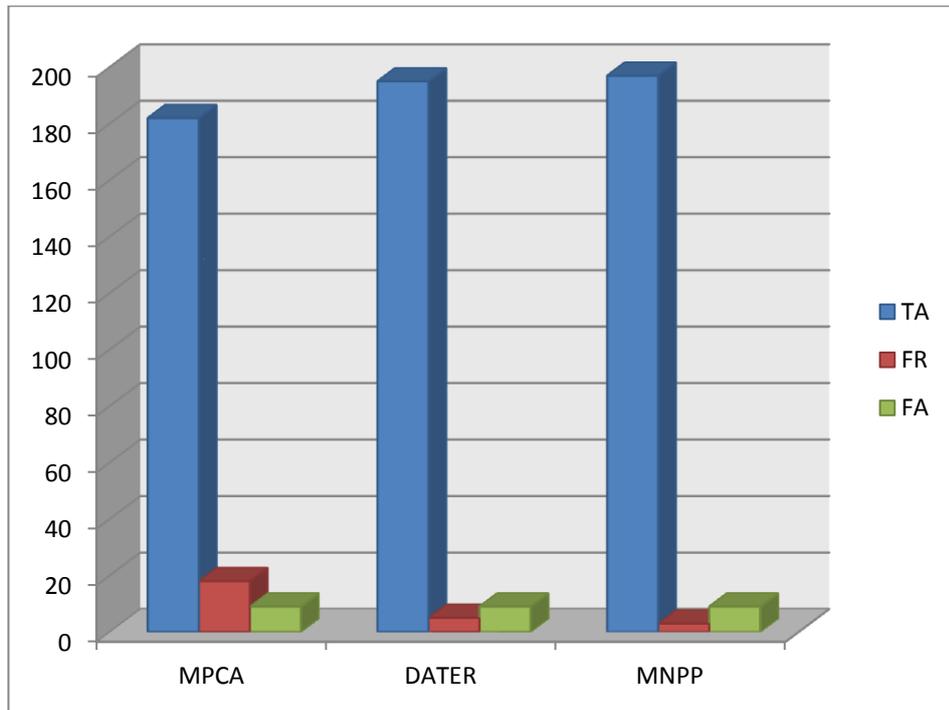


Figure 5.22: Comparisons of TA, FR, and FA values for the MPCA, DATER and proposed MNPP method with the AR colour database and varying facial expressions.

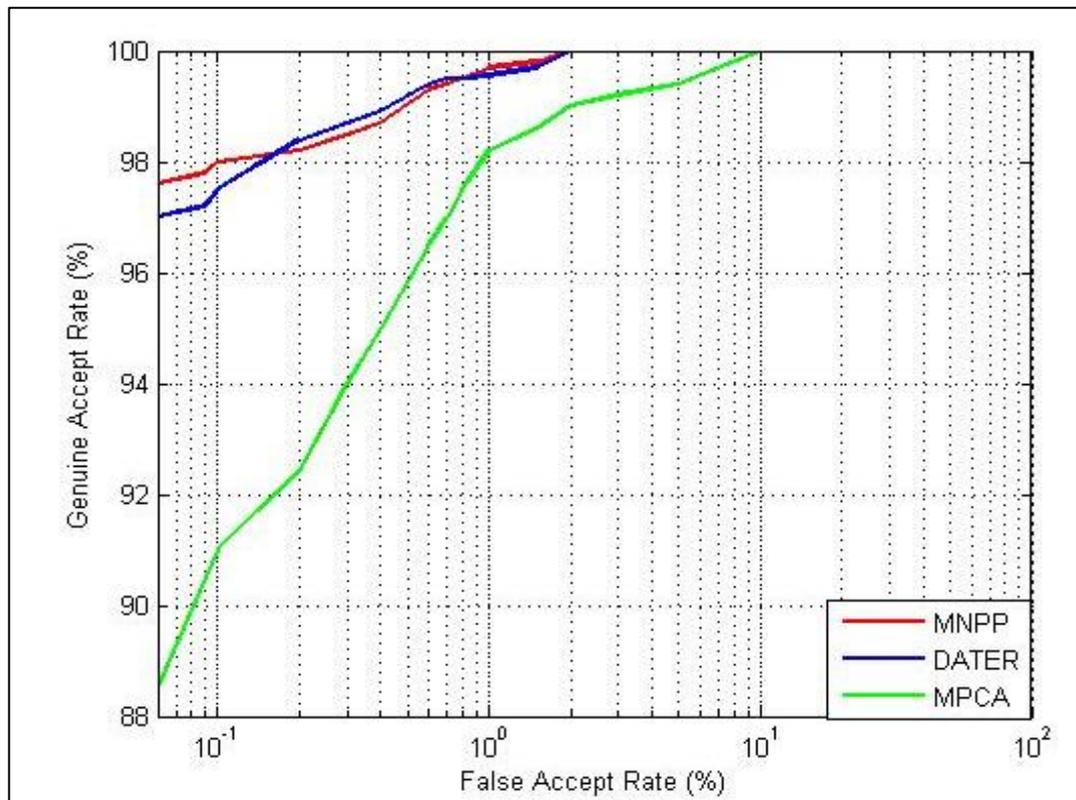


Figure 5.23: Comparisons of ROC curves of the of the MPCA, DATER and the proposed method MNPP on AR colour database under varying facial expressions

Table 5.17: Values of EER and GAR for the MPCA, DATER and proposed MNPP method with the AR colour database and varying lighting conditions

Method	EER	GAR at the lowest FAR (0.1)
MPCA	2.95	54.5
MLDA	1.75	79.0
MNPP	0.27	98.4

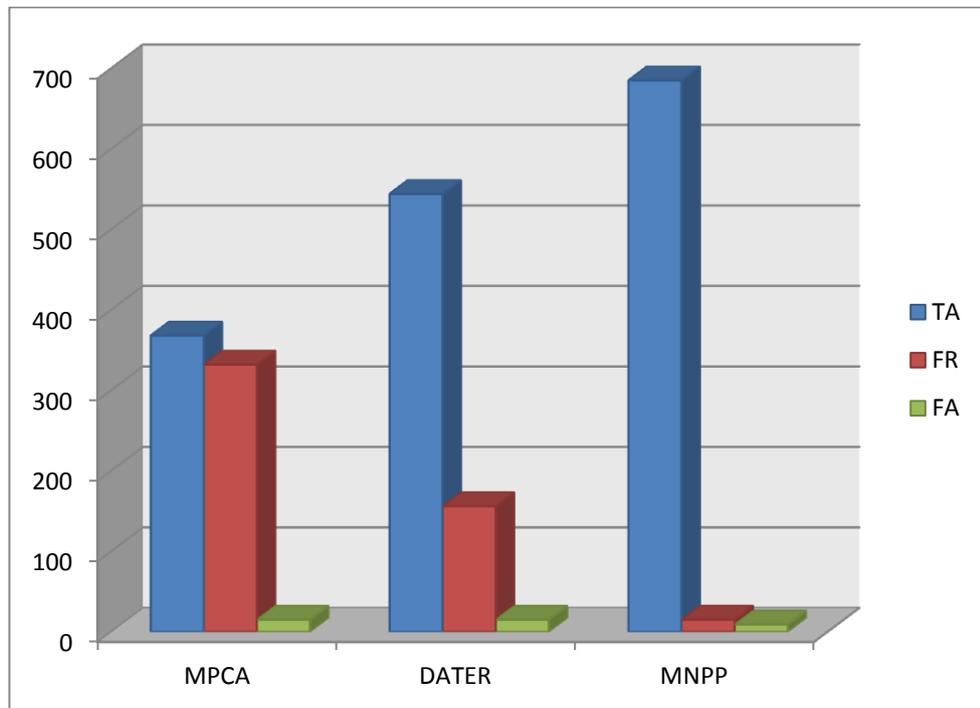


Figure 5.24: Comparisons of TA, FR, and FA values for the MPCA, DATER and proposed MNPP method with the AR colour database and varying lighting conditions

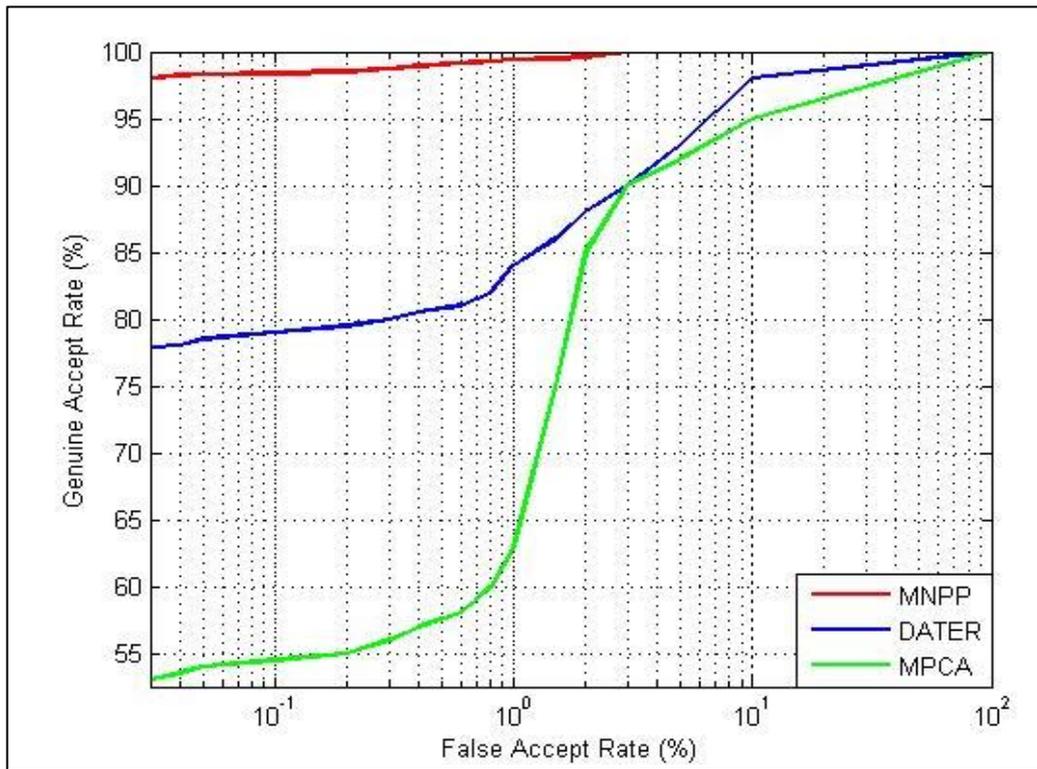


Figure 5.25: Comparisons of the ROC curves for MPCA, DATER and proposed MNPP method with the AR colour database and varying lighting conditions.

Table 5.18: Values of EER and GAR for the MPCA, DATER and proposed MNPP method with the AR colour database and varying occlusion

Method	EER	GAR at the lowest FAR (0.5)
MPCA	8.63	21.2
MLDA	3.60	59.4
MNPP	1.45	72.9

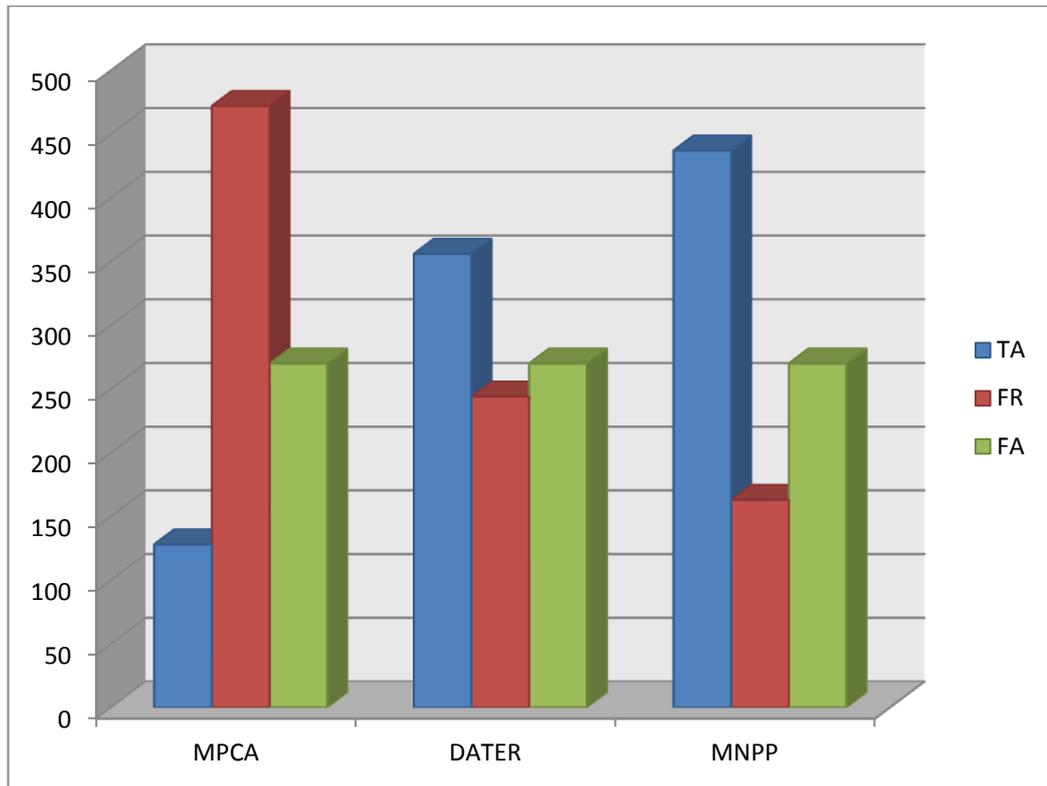


Figure 5.26: Comparisons of TA, FR, and FA values for the MPCA, DATER and proposed MNPP method with AR colour database and varying occlusion

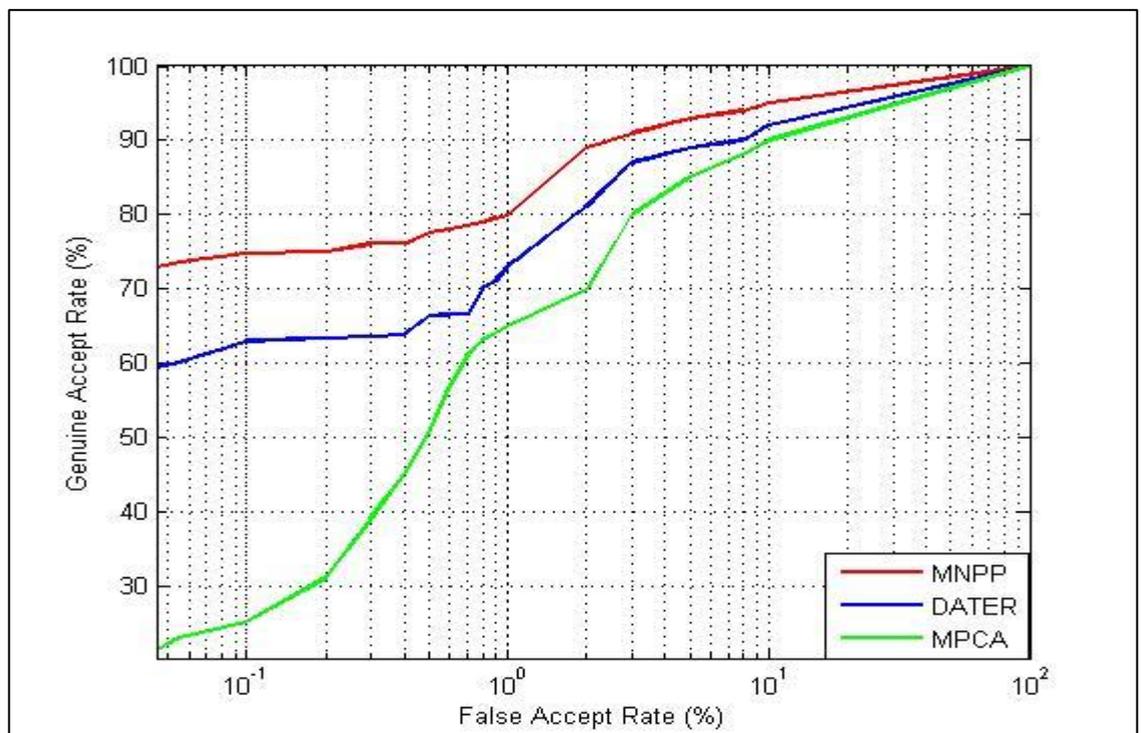


Figure 5.27: Comparisons of the ROC curves for the MPCA, DATER and proposed MNPP method with the AR colour database and varying occlusion

Table 5.19: Values of EER and GAR for the MPCA, DATER and proposed MNPP method with the FERET colour database.

Method	EER	GAR at the lowest FAR (0.1)
MPCA	2.93	74.0
MLDA	1.75	83.8
MNPP	0.90	89.2

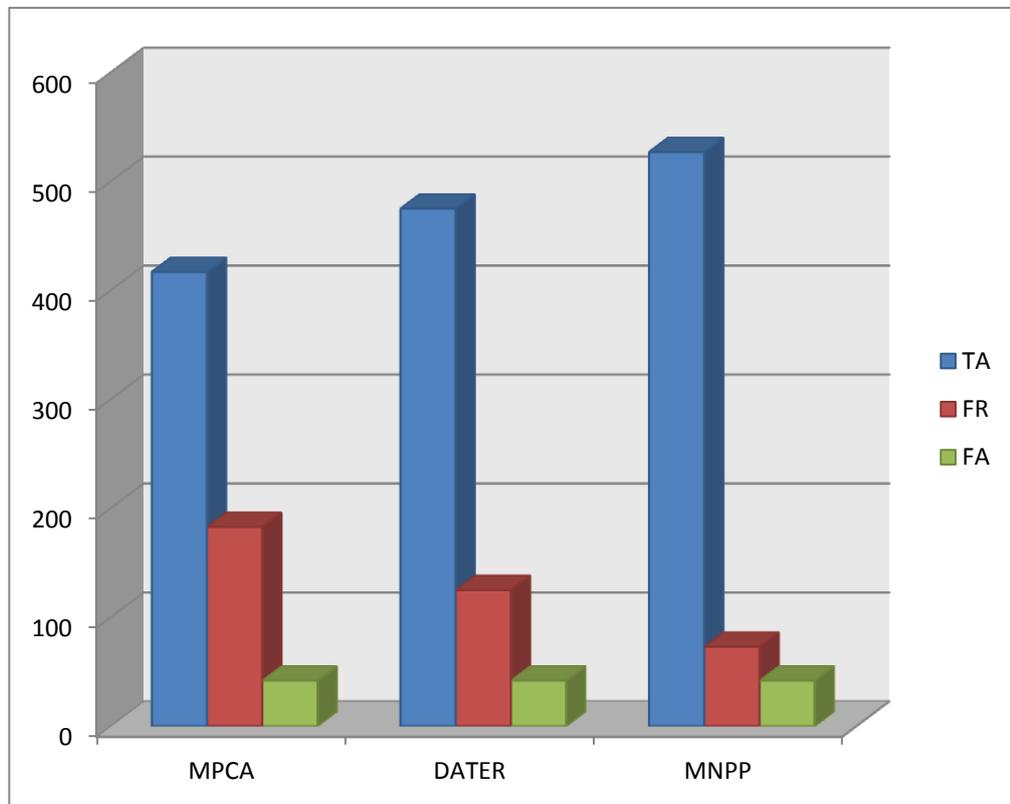


Figure 5.28: Comparisons of TA, FR, and FA values for the MPCA, DATER and proposed MNPP method with the FERET colour database.

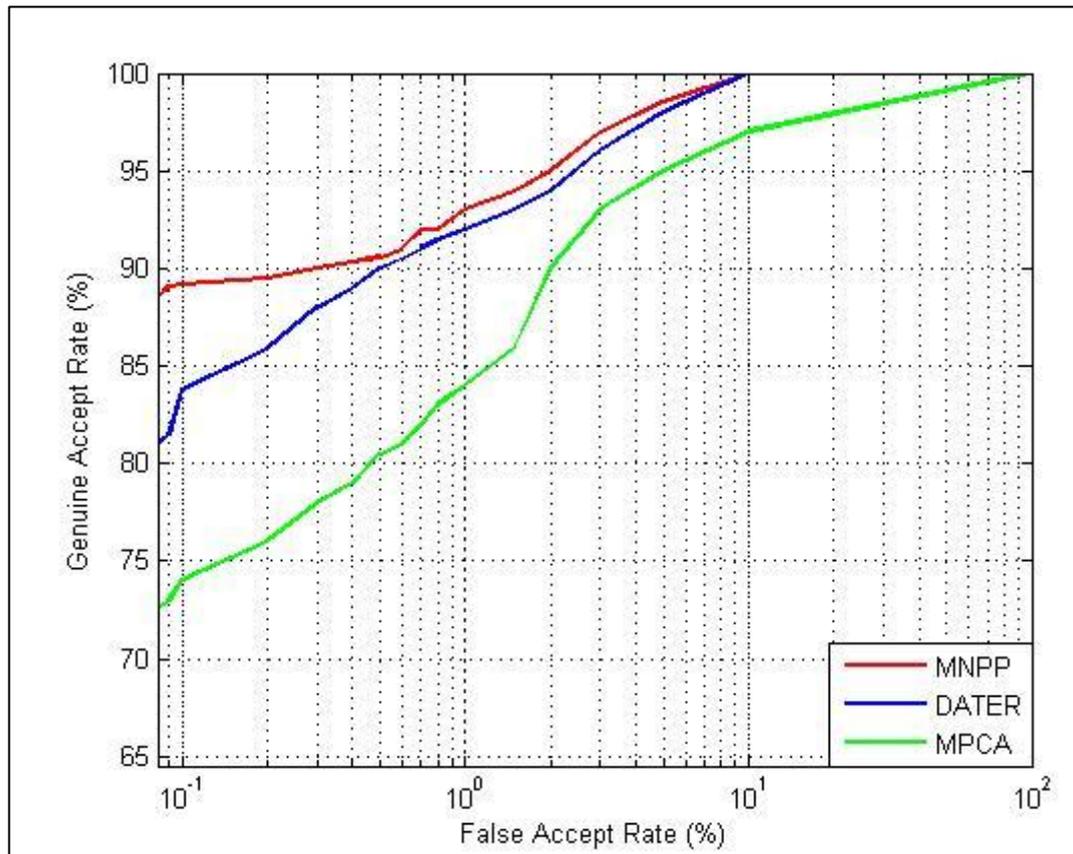


Figure 5.29: Comparisons of the ROC curves for the MPCA, DATER and proposed MNPP method with the FERET colour database under

It is clear from the results of these extensive experiments represented in the above figures and tables that the supervised MNPP outperforms all other methods in both identification and verification modes. Detailed discussions of these results are given in Section 5.6.5.

#### 5.6.4 Dimensionality selection

In this study, a number of comparisons have been conducted to show the importance of feature selection for 1-D, 2-D and  $N$ -D versions, of all of the methods tested in this Chapter (PCA, LDA, LPP, and NPP) incorporating data from the three databases (ORL, AR, and FERET).

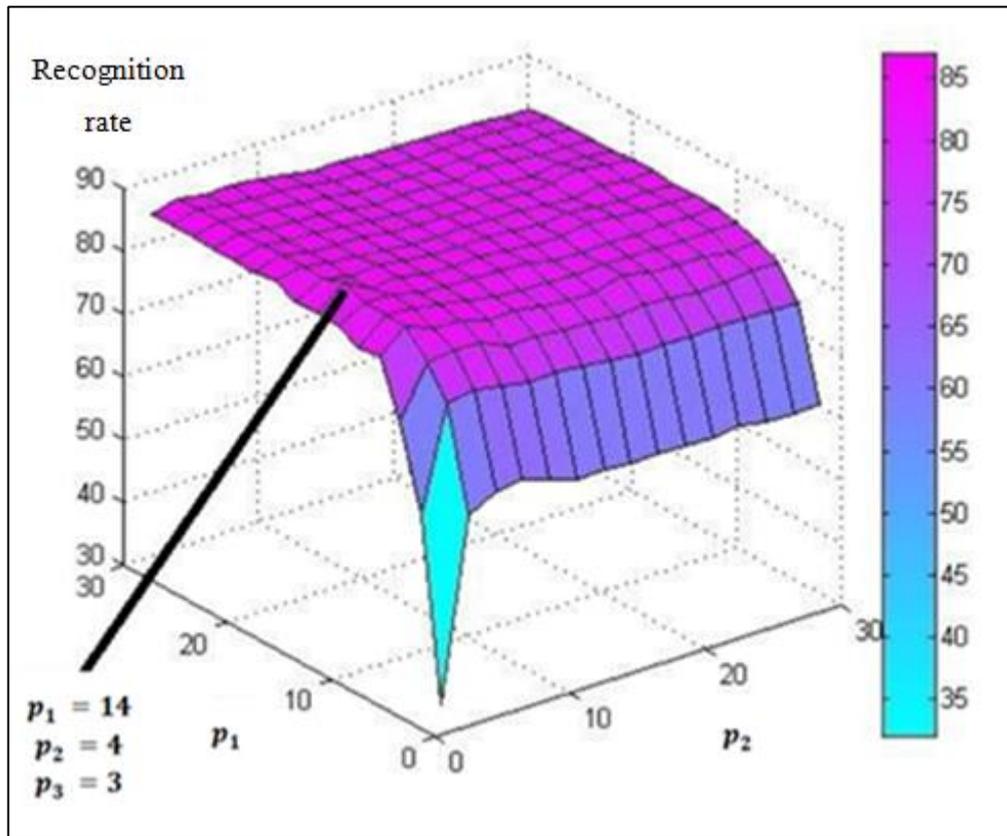


Figure 5.30: Number of eigenvectors selected against detection recognition rates with the colour FERET database.

For PCA, LDA, LPP and NPP, experiments were run to compute the error rate while varying  $p$  from 1 to 200. For 2DPCA, 2DLDA, 2DLPP and 2DNPP, the error rate was also tested using a different number of eigenvectors  $p_1$  from 2 through 30. For MPCA, DATER, ND-TSNE and MNPP several trials were executed by varying the number of eigenvectors  $p_1$  and  $p_2$  from 2 to 30, while varying  $p_3$  from 1 to 3. The optimum subset of eigenvectors was then selected so that the total error rate was minimized for all three datasets. Figure 5.30 illustrates the recognition rate with the FERET database for the MNPP approach against the number of selected eigenvectors.

It is clear that the MNPP achieved an average recognition rate of 85.9% (error rate of 14.1%) when  $p_1 = 14$ ,  $p_2 = 4$  and  $p_3 = 2$ . This study proves that the MNPP achieves the smallest error rate and highest recognition performance compared to the other methods using the smallest number of extracted features.

### 5.6.5 Discussion

Having performed a large number of experiments with facial image databases to evaluate the performance of the proposed MNPP method in terms of the two modes of facial recognition biometrics systems of identification and verification, a number of observations can be highlighted from the results:

1. From the experiments on original grey (2-D tensor) and colour (3-D tensor) level features, the identification results clearly demonstrate that the supervised MNPP has the potential to outperform all the linear and multilinear subspace learning methods in all cases and with different databases. Moreover, the verification results indicate that the MNPP achieved acceptable rates more effectively than the NPP, 2DNPP, MPCA, and DATER methods with respect to the EER and GAR. Its correct recognition rates (CRR), equal error rates (EER), genuine acceptance rates (GAR), and true acceptance (TA), false rejection (FR), and false acceptance (FA) scores are better than those of all the other methods tested across all experiments. This indicates that features extracted directly by the supervised MNPP from the tensorial facial data for each subject are much more effective in classification.
2. The proposed unsupervised and supervised MNPP approach overcomes the limitations of one- and two-dimensional methods in the reduction of dimensionality, as well as that of the number of useful features extracted. From the experiments, it can be concluded that the supervised MNPP method can effectively capture most of the local structure and discriminative information, with only a small number of features, as opposed to the other methods where more features have to be provided in order to achieve a better recognition rate. Therefore, it can be seen that the supervised MNPP method outperforms the PCA, LDA, NPP, LPP, 2DPCA, 2DLPP, 2DLDA and 2DNPP methods in all tests in terms of error rate, with all the three databases, as well as improving recognition. These other methods implement compression and preserve relationships only in one direction, while the MNPP performs reduction while preserving the hidden relationships in both column and row directions.
3. It is interesting to note that both the 2DLDA and the DATER methods work better than the unsupervised MNPP method only with the AR dataset sample where facial expressions vary. This is because the unsupervised MNPP preserves

only local geometric information whereas the LDA forms (LDA, 2DLDA and DATER) preserve discriminative information which is important in recognizing the same faces with different facial expressions. Therefore, using the supervised MNPP and taking into consideration the label information about the training faces further improved face recognition performance so that the proposed method now outperforms both 2DLDA and DATER as well. This is because the supervised MNPP preserves the intrinsic geometry in addition to the discriminative information in the image space. On the other hand, the performance of LDA forms was much worse than both supervised and unsupervised MNPP with the AR data in varying lighting condition and occlusions as well. This indicates that, when faces contain local variations such as those resulting from different lighting conditions and occlusions, the MNPP works much better than all of the other comparable methods, including DATER, while the LDA forms result in lower recognition performance. This is because the MNPP preserves the local neighborhood structure and intrinsic geometry of the image space, and so it generalizes local features which are less sensitive to variation illumination. Thus the faces can be more easily distinguished and recognized. Whereas LDA forms only preserve the global geometry and discriminative information in the data space, and the lack of local geometry descriptions causes lower recognition performance.

4. From Tables 5.7 and 5.8, it can be seen that all methods are unable to adequately process occluded parts of the face, which is the most difficult task in recognition. This is due to the fact that images incorporating mouth occlusions contain the least amount of significant identity information, thus resulting in lower values of similarity scores. This is especially evident with the PCA, 2DPCA and MPCA methods, which have the relative disadvantage of using a global similarity score. However, a possible way to improve recognition performance is by using local approaches, such as ND-TSNE, DATER and MNPP, which perform much better than the MPCA. Therefore, the ability of the proposed method to recognize faces when occlusions are present with high accuracy levels than other methods makes it potentially very useful in various face applications.
5. Comparing the performance of biometric systems at the lowest false acceptance rates depends on the EER and GAR. The best system has the smallest EER and largest GAR. Tables 5.11-5.19 prove that the MNPP system can be considered

more accurate than the other systems such as NPP, 2DNPP, MPCA, DATER, and ND-TSNE at a specified FAR value of 0.1%) because it also has lowest EER values. Furthermore, all the ROC graphs shown in Figures 5.13, 5.15, 5.17, 5.19 5.20, 5.23, 5.25, 5.27 and 5.29 show that the proposed MNPP method for facial recognition verification provides consistent performance with respect to GAR. its performance is much better than those of the competition algorithm at a lowest FAR value of 0.1% due to it also having the highest GAR values.

6. The verification results clearly show that reducing the value of EER resulted in a decrease in the number of incorrectly classified images. Therefore, the proposed MNPP method has the most effective verification system which produces the lowest EER compared with methods.
7. Compared with the MPCA and the DATER, both of which preserve the global structure of data samples at the cost of losing essential information about the local geometry of the neighbors, the MNPP protects the local manifold structure. This is more important than the global Euclidean structure, especially in facial recognition and data reduction problems. Moreover, the supervised MNPP preserves all of the intrinsic geometry, local neighborhood and discriminative information in the image dataspace.
8. Furthermore, the DATER and ND-TSNE are implemented in a supervised way, with only additional information about class labels available. However, the MNPP and 2DNPP can perform in both supervised and unsupervised forms without needing any additional parameters.
9. The MNPP does not need any parameter selection during the construction of the neighborhood weighting affinity matrix, whereas the LPP, 2DLPP and ND-TSNE methods are very sensitive to the selection of the width of a Gaussian envelope.
10. The comparisons conducted in this study prove that the MNPP exhibits reasonable and acceptable performance against the baseline PCA, LDA, LPP and NPP solutions and their existing extensions. Thus, the proposed method has practical relevance in security-related applications such as biometric authentication and surveillance [6].
11. Compared with traditional methods such as PCA, LDA, LPP and NPP, our proposed MNPP method effectively avoids the dimensionality dilemma and alleviates the problem of small sample size.

## 5.7 Summary

In this chapter, a novel supervised and unsupervised multilinear neighborhood preserving projection (MNPP) method has been proposed for the extraction of discriminative features by extending the original NPP to its multilinear form, which is now suitable for use in face recognition and security biometrics applications.

Theoretical analysis of the MNPP has also been presented in this chapter. The proposed method has been validated using three different benchmark facial databases, namely, the ORL, AR and FERET. The experimental results for the two face recognition biometric modes of identification and verification show that error rates with the MNPP proposed method have been significantly reduced in comparison to those of conventional subspace analysis methods. In addition, the unsupervised MNPP performs the best amongst most of the methods evaluated, which validates the effectiveness of natural tensorial data. Moreover, results for the supervised MNPP prove that it outperforms all of the standard approaches across all experiments with respect to error rate (ERR), genuine acceptance rate (GAR), and equal error rates (EER). Therefore, the comparative evaluation demonstrates that the MNPP is a robust and effective recognition method for tensor objects.

# Chapter 6:

## Conclusion and Future Work

This chapter summarises the research work presented in the thesis. The key contributions of the work in the field of facial recognition are highlighted. Nevertheless, there still remain open questions to be addressed in the future. Thus, potential future directions for research are towards the aim of achieving more efficient MSL methods also outlined.

### 6.1 Summary and Major Contributions of the Research

The search for a robust automatic facial recognition system is a challenging problem due to the large variability of the appearance of human faces, insufficient data samples, and the complexity of pattern distribution. Linear subspace learning (LSL) techniques attempt to map high-dimensional data into a low-dimensional subspace in order to improve the task of automatic recognition. However, classical subspace learning methods such as NPP are linear methods. Facial samples thus need to be reshaped from their natural tensorial representations into very high-dimensional vectors. This reshaping breaks the natural structure of and correlations within the original tensorial data, as well as involving high computational costs and the need to estimate a huge number of parameters. Unlike LSL, multilinear subspace learning (MSL) methods handle facial objects in their original tensorial form. Also, they have the capacity to learn useful and more compact representations which can result in recognition. Therefore, this thesis has focused on investigating the multilinear subspace learning approach to appearance-based human facial recognition. It has contributed to the development of multilinear subspace learning algorithms.

The work in this thesis has fulfilled the aims of the research set out in Chapter 1. Chapter 2 presented a literature review of LSL techniques. Some popular one-dimensional subspace learning methods which map input images into high-dimensional subspace are explained. The problems of and solutions to two- and bidirectional-dimensional subspace learning which operate directly on 2D-images are illustrated.

Chapter 3 provided an overview of MSL techniques for the dimensionality reduction of tensorial data taken directly from their natural representations. Multilinear algebra, projection, and prominent existing algorithms are reviewed. However, in addition to existing linear and multilinear subspace learning approaches, some bidirectional two-dimensional and multilinear subspace learning approaches that have not yet been studied.

Chapter 4 presented two new facial recognition algorithms. The first is called bidirectional two-dimensional neighborhood preserving projection (B2DNPP) feature extraction. It addresses the problem of the bidirectional two-dimensional method by expanding the two-dimensional neighborhood preserving projection (2DNPP) method into its bidirectional two-dimensional form in order to improve the performance of the unidirectional approach [36]. Whereas the 2DNPP performs compression in the direction of the image columns only, the proposed B2DNPP method works in both row- and column- directions in order to preserve the spatial information and correlation between image rows and columns. The B2DNPP builds only one weight matrix from the nearest neighbor affinity graph, which is then used to sequentially extract right- and left- projection matrices. These two projection matrices are used to transform the facial images into low-dimensional space where the facial recognition task is much easier. As a result, B2DNPP has many important advantages over the 2DNPP method. Firstly, it shows the hidden relationships among the image row and column vectors. Secondly, it enhances the recognition performance of the 2DNPP method. Furthermore, image feature extraction is simpler and it is easier to evaluate the relationships within the image matrix accurately, so that less computational time is required.

The second approach presented in Chapter 4 is called C-B2DNPP, which is a bidirectional two-dimensional neighborhood preserving projection method based the curvelet decomposition of human facial image data intended to give better recognition performance. In this method, the curvelet sub-bands that demonstrate high variation are selected as initial image features. The selecting method of curvelet coefficients depends on the amount of variation that each possesses. Thus, an approximated coefficient along with four coefficients from sub-band 2 are selected, which further improves recognition performance. The B2DNPP algorithm is applied to the selected coefficients to preserve the local structure of the data manifold and generate distinctive feature vectors. This framework uses an extreme learning machine (ELM) classifier for classification, which

significantly enhances classification rate. Extensive experiments have been performed to evaluate the performance of B2DNPE and C-B2DNPP using the three benchmark human facial databases ORL, AR, and FERET. It is proven that C-B2DNPP outperforms 2DNPP and all existing bidirectional 2-D methods on the various databases with respect to recognition rate, computational time, and the numbers of selected eigenvectors.

Chapter 5 naturally extended the standard 1-D NPP to its multilinear form by developing a novel biometric face recognition approach based on the multilinear neighborhood preserving projection algorithm (MNPP), which operates directly on the natural tensorial data rather than vectors or matrices. It enhances face recognition performance compared to the B2DNPP which works only on 2D matrices and overcomes the limitations of existing MSL algorithms. Traditional approaches such as NPP and 2DNPP, and even B2DNPP, derive only one or even two subspaces, while the MNPP method obtains multiple interrelated subspaces by unfolding the tensorial data over different tensor directions. The number of subspaces derived by MNPP depends on the order of the tensorial data space. In general, the MNPP involves three main steps: 1) pre-processing, the tensorial samples cropped and rotated here, and the cropped objects are resized to ensure that all tensors are of the same size; 2) the MNPP framework for feature extraction preserves the relationships of the natural tensorial data which is the most significant part of this system; 3) classification, which that recognizes faces by classifying the MNPP feature vectors. The proposed MNPP method has the following characteristics:

1. It solves the problem of supervised and unsupervised dimensionality reduction for embedded manifolds, and thus allows a natural representation of multidimensional images in 2-D, 3-D or higher-order tensors.
2. It derives a multiple interrelated lower-dimensional dataspace for feature extraction and selection by unfolding the tensor across all tensor dimensions. It learns the subspaces iteratively by unfolding the tensor along different tensor directions.
3. It preserves the local structure of data samples in addition to the discriminative information in the feature space using natural 3-D and even higher-order tensorial data in order to reveal the essential structures for data analysis, without requiring the data to be vectorized.

4. Compared with traditional methods such as PCA, LDA, LPP and NPP, the proposed method effectively avoids the problems of small sample size (SSS) and the dimensionality dilemma.

The performance of the MNPP method has been compared with that of conventional methods such as PCA [15], LDA [14], LPP [31], NPP [33], 2DPCA [47], 2DLDA [48], 2DLPP [49], 2DNPP [50], MPCA [46], DATER[26], and ND-TSNE [44]. The proposed method has been validated by several kinds of analysis using three different benchmark facial databases, namely ORL, AR and FERET. The results of the experimental studies for the two face biometric recognition modes of identification and verification show that the error rates with the proposed MNPP method are much lower than those of the conventional method of subspace analysis. The unsupervised MNPP performs the best amongst most of the evaluated methods, while the comparative evaluation demonstrates that the supervised MNPP is a robust and effective recognition method for tensor objects since it outperforms all of the standard approaches across all experiments with respect to error rate (ERR), genuine acceptance rate (GAR), and equal error rates (EER).

## **6.2 Future Research**

Although many facial recognition problems in multilinear subspace learning have been addressed, the MSL field still has many unresolved problems which need to be considered. Therefore, this section presents some research topics for future investigation with the aim of developing novel applications where the proposed MNPP method can be used, and in designing multilinear subspace learning solutions.

1. Combine MNPP features with other features

Feature fusion has become one of the most effective methods used to improve recognition performance. The combination in Chapter 4 of B2DNPP with the curvelet transform and extreme learning machine (ELM) for single-hidden layer feedforward neural networks has shown promising results for facial recognition. Therefore, it would be interesting to investigate the combination of the MNPP with the curvelet transform and ELM classifier. Moreover, the 2DLDA and DATER algorithms gave the highest recognition rates after the MNPP in the case of varying facial expression, so it would be worth investigating the

integration of the feature vectors extracted by MNPP with other features extracted by traditional linear subspace methods such as LDA, (2D)2FLD, and DATER in order to further reduce dimensionality and extract the most discriminant information for better separation.

2. Parallel model for face recognition

a) Any facial image can be analyzed using different algorithm in either the frequency or spatial domains. Each domain describes the image in a different way. Therefore, it would be interesting to consider whether or not the utilization of spatial features extracted from the MNPP and the features extracted from one of the frequency domain algorithms in a parallel model may provide better performance. Frequency features are extracted using for example, the discrete cosine transform (DCT), and both types of features can be extracted in a parallel method as described below in Figure 6.1. Then, these multiple features can be integrating in vectors which are later fed to a classifier.

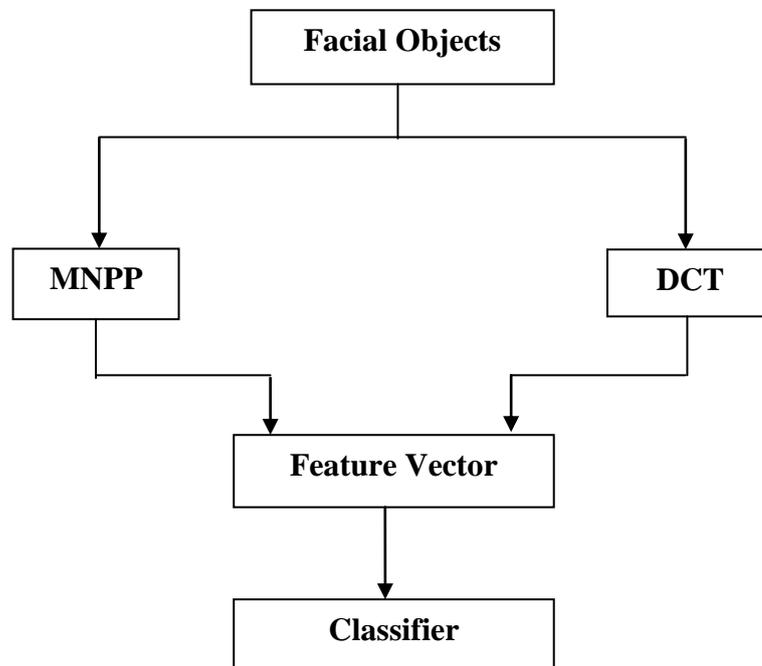


Figure 6.1: Parallel model for facial recognition

- b) Furthermore in the same parallel way it would be worthwhile to determine if the integration of MNPP features with those of existing multilinear algorithms such as MPCA, DATER, and ND-TSNE has the ability to improve recognition rates, as shown in Figure 6.2. Then testing this new method to be executed in parallel using one of existing parallel machines such as multicores for gaining more speed and efficiency.

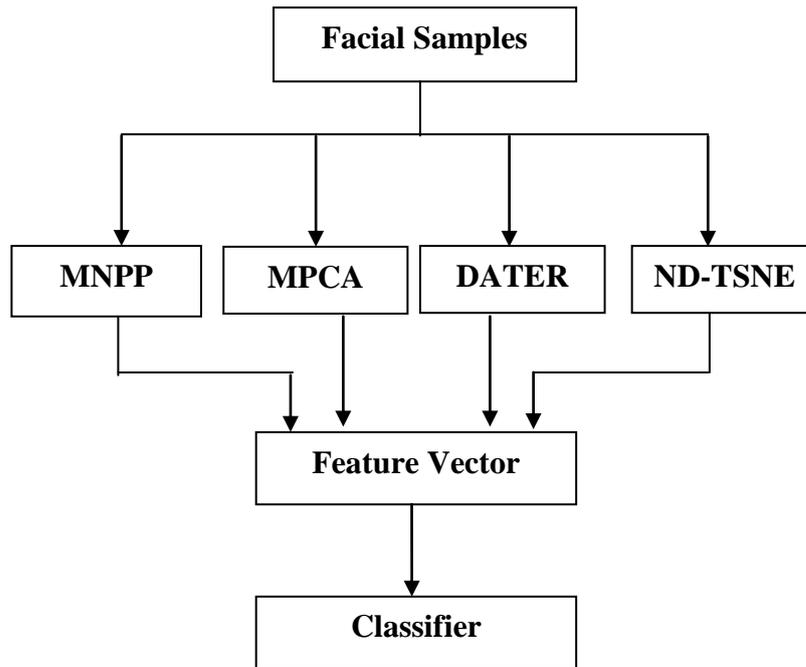


Figure 6.2: Parallel multidimensional model for facial recognition

### 3. V-MNPP

The solution in the MNPP approach is based on tensor-to-tensor projection. It would be interesting to derive a multilinear solution for the NPP method based on tensor-to-vector projection (V-MNPP). In this type of projection, the number of features that can be extracted will be less than the smallest dimension. Therefore, this method may be more suitable for recognition tasks where only a small number of features of needed.

### 4. Using MNPP with other applications

As our lifestyles become increasingly dependent on digital technology, it is believed that the increased efficiency of this technology should be able to enhance our quality of life and the security of our information. Therefore, it would be interesting to design an authentication system that provides easy and safe access to computers without requiring our attention, time and energy. Such a system should not be intrusive and would need no specific material that not already available in classical computers. Also, it should be highly acceptable to users as well as exploiting low-cost biometric modalities. Therefore, the design of an authentication system that detects and recognizes persons depending on video-based facial recognition via webcam could be investigated as a new application for the MNPP to analyses video sequences. Interesting future applications of the proposed MNPP method may include applying it to other biometrics such as fingerprints and gait. Beside biometric facial recognition applications, many other computer vision and pattern recognition applications use tensorial data too. It is planned that some of these applications will be explore, such as medical image analysis and for use in daily activities in human-computer interaction (HCI) using MNPP.

#### 5. Hybrid biometric system

Research into multimodal biometrics systems is able to overcome some of the drawbacks of uni-biometric systems by grouping multiple sources of information together. These systems use at least two biometrics characteristics, including physiological or behavioural variables, in order to recognize individuals. Thus, it would be interesting to use the MNPP in such multimodal biometrics systems which combine for example the features of facial images, fingerprints and/or palmprints.

## REFERENCES

- [1] Jain, A.K.; Bolle, R.; Pankanti, S., eds, *Biometrics: Personal Identification in Networked Society*: Kluwer Academic Publications, 1999.
- [2] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell, "Face recognition by humans: Nineteen results all computer vision researchers should know about," *Proceedings of the IEEE*, vol. 94, pp. 1948-1962, 2006.
- [3] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, pp. 4-20, Jan. 2004.
- [4] Available: <http://news.bbc.co.uk/1/hi/england/wear/3115428.stm>
- [5] L.H. Simon and M.E. Pheneger, "American Civil Liberties Union (ACLU) call for Public Hearings on Tampa's "Snooper Bowl" Video Surveillance," 2001, Available on line on: <http://www.aclu.org/technology-and-liberty/aclu-calls-public-hearings-tampas-snooper-bowl-video-surveillance>.
- [6] R. Chellappa, A. Roy-Chowdhury, and S. Zhou, *Recognition of Humans and Their Activities Using Video*: Morgan & Claypool Publishers, San Rafael, California, 2005.
- [7] P. S. Penev and J. J. Atick, "Local feature analysis: a general statistical theory for object representation," *Network: Computation in Neural Systems*, vol. 7, pp. 477-500, 1996.
- [8] B. S. Manjunath and R. Chellappa, "A feature based approach to face recognition," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 373-378, 1992.
- [9] L. Wiskott, J.-M. Fellous, N. Kruger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19(7), pp. 775-779, 1997.
- [10] A. Hadid T. Ahonen and M. Pietikainen, "Face recognition with local binary patterns," *Proc. European Conference on Computer Vision*, pp. 469-481, 2004.
- [11] S. Arca, P. Campadelli, and R. Lanzarotti, "A face recognition system based on local feature analysis," *Proc. Audio- and Video-Based Biometric Person Authentication*, pp. 182-189, 2003.
- [12] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local gabor binary pattern histogram sequence (LGBPHS): A novel non-statistical model for face

- representation and recognition," *Proc. IEEE International Conference on Computer Vision and Image Understanding*, pp. 786-791, 2005.
- [13] N. Vaswani and R. Chellappa, "Principal components null space analysis for image and video classification," *IEEE Transactions on Image Processing*, vol. 15(7), pp. 1816-1830, 2006.
- [14] J. P. Hespanha P. N. Belhumeur and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19(7), pp. 711-720, 1997.
- [15] M. Turk and A. Pentland, "Eigenfaces for recognition," *Cognitive Neuroscience*, vol. 3, pp. 72-86, 1991.
- [16] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12(1), pp. 103-108, 1990.
- [17] X. Lu, "Image analysis for face recognition," 2003. [Online]. Available: <http://www.face-rec.org/interesting-papers/General/ImAna4FacRcglu.pdf>.
- [18] M. S. Nixon and J. N. Carter, "Automatic recognition by gait," *Proceedings of the IEEE*, vol. 94, pp. 2013-2024, 2006.
- [19] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1042-1052, Oct. 1993.
- [20] W. Zhao, R. Chellappa, A. Rosenfeld, and P. Phillips, "Face recognition: A literature survey," *ACM Computing Surveys*, pp. 399-458, 2003.
- [21] J. Lu, K. N. Plataniotis, A. N. Venetsanopoulos, and S. Z. Li, "Ensemble-based discriminant learning with boosting for face recognition," *IEEE Transactions on Circuits and Systems for Video Technology Neural Networks*, vol. 17, pp. 166-178, Jan. 2006.
- [22] X. He, D. Cai, and P. Niyogi, "Tensor subspace analysis," *Advances in Neural Information Processing Systems 18 (NIPS)*, 2005. [Online]. Available: [http://books.nips.cc/papers/\\_les/nips18/NIPS2005\\_0249.pdf](http://books.nips.cc/papers/_les/nips18/NIPS2005_0249.pdf).
- [23] W. H. Greub, *Multilinear Algebra*: Berlin: Springer-Verlag, 1967.
- [24] L. P. Lebedev and M. J. Cloud, *Tensor Analysis*: World Scientific, 2003.
- [25] S. Lang, *Algebra*: Reading: Addison Wesley, 1984.

**References** .....

---

- [26] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. Zhang, "Multilinear discriminant analysis for face recognition," *IEEE Transactions on Image Processing*, vol. 16, pp. 212-220, Jan. 2007.
- [27] G. Shakhnarovich, B. Moghaddam, "Face recognition in subspaces," *Handbook of Face Recognition*, vol. S. Z. Li and A. K. Jain, Eds. Springer-Verlag, pp. 141-168, 2004.
- [28] J. Zhang, S. Z. Li, and J. Wang, *Manifold learning and applications in recognition*: Y. P. Tan, K. H. Yap, and L. Wang, Eds. Springer-Verlag, 2004.
- [29] M. H. C. Law and A. K. Jain, "Incremental nonlinear dimensionality reduction by manifold learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 377-391, Mar. 2006.
- [30] X. He, P. Niyogi, "Locality Preserving Projections," *Proceedings Conference of Advances in Neural Information Processing Systems*, 2003.
- [31] X. He, S. Yan, Y. Hu, P. Niyogi, H. Zhang g, "Face recognition using laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1-13, 2005.
- [32] X.F. He, D. Cai, S.C. Yan, and H.J. Zhang, "Neighborhood Preserving Embedding," *IEEE Conf. on ICCV'05*, vol. 2, pp. 1208-1213, 2005.
- [33] Y. Pang, L. Zhang, Z. Liu, "Neighborhood preserving projections (NPP): A novel linear dimension reduction method," *Lecture Notes in Computer Science*, vol. 3644, pp. 117-125, 2005.
- [34] D. Tao, X. Li, X.Wu, and S. J. Maybank, "Elapsed time in human gait recognition: A new approach," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 177 - 180, Apr. 2006.
- [35] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and gabor features for gait recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1700-1715, Oct. 2007.
- [36] D. Xu, S. Yan, L. Zhang, H.-J. Zhang, Z. Liu, and H.-Y. Shum, "Concurrent subspaces analysis," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. II, June 2005.
- [37] G. Hua, P. A. Viola, and S. M. Drucker, "Face recognition using discriminatively trained orthogonal rank one tensor projections," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-7, June 2007.

**References** .....

---

- [38] Y. Wang and S. Gong, "Tensor discriminant analysis for view-based object recognition," *Proc. International Conference on Pattern Recognition*, vol. 3, pp. 33 - 36, August 2006.
- [39] J. Ye, R. Janardan, and Q. Li, "GPCA: An efficient dimension reduction scheme for image compression and retrieval," *The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 354-363, 2004.
- [40] J. Ye, R. Janardan, and Q. Li, "Generalized low rank approximations of matrices," *Machine Learning*, vol. 61, pp. 167-191, 2005.
- [41] A. Shashua and A. Levin, "Linear image coding for regression and classification using the tensor-rank principle," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. I, pp. 42-49, 2001.
- [42] S. Yan, D. Xu, B. Zhang, H. J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 40-51, Jan. 2007.
- [43] G. Dai and D. Y. Yeung, "Tensor embedding methods," *Proc. Twenty-First National Conference on Artificial Intelligence*, pp. 330-335, Jul. 2006.
- [44] X. Han, X. Qiao, and Y. Chen, "Multilinear Supervised Neighborhood Embedding with Local Descriptor Tensor for Face Recognition," *IEICE Trans. INT.&SYST.*, vol. E94-D, pp. 158-161, 2011.
- [45] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear Subspace Analysis for Image Ensembles," *IEEE Conference on Computer Vision and Pattern Recognition, Madison, WI*, vol. 2, pp. 93-99, 2003.
- [46] H.P. Lu, K.N. Plataniotis, and A.N. Venetsanopoulos, "MPCA: multilinear principal component analysis of tensor objects," *IEEE T. Neural. Networ.*, vol. 19, pp. 18-39, 2008.
- [47] J. Yang, D. Zhang, A. F. Frangi, "Two-dimensional PCA: A new approach to appearance based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, 2004.
- [48] M. Li, B. Yuan, "2D-LDA: A statistical linear discriminant analysis for image matrix. PatternRecognition Letters," *Pattern Recognition Letters*, vol. 26, pp. 527-532 2005.
- [49] S. Chen, H. Zhao, M. Kong, B. Luo., "2D-LPP: A two-dimensional extension of locality preserving projections," *Neurocomputing*, vol. 70, pp. 912-921, 2007.

*References* .....

---

- [50] H. Zhang, Q.M. Wu, T.W. Chow, and M. Zhao, "A two-dimensional Neighborhood Preserving Projection for appearance- based face recognition," *Pattern Recognition*, vol. 45, pp. 1866-1876, 2012.
- [51] D. Zhang, Z. H. Zhou, "(2D) 2PCA: Two-directional two-dimensional pca for efficient face representation and recognition," *Neurocomputing*, vol. 69, pp. 224-231, 2005.
- [52] P. Nagabhushan, D. S. Guru, B. H. Shekar, "(2D) 2 FLD: An efficient approach for appearance based object recognition," *Neurocomputing*, vol. 69, pp. 934-940, 2006.
- [53] S. B. Chen, B. Luo, G. P. Hu, R. H. Wang, "Bilateral two-dimensional locality preserving projections," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 601-604, 2007.
- [54] A.A. Mohammed, R.Minhas, Q.M. Jonathan Wu and M.A. Sid-Ahmed, "Human Face Recognition based on Multidimensional PCA and Extreme Learning Machine," *Pattern Recognition*, vol. 44(10-11), pp. 2588-2597, October-November 2011.
- [55] Haro, M. Flickner, and I. Essa, "Detecting and tracking eyes by using their physiological properties, dynamics, and appearance," *Proc. IEEE Conf.. Computer Vision and Pattern Recognition*, vol. 1, pp. 163-168, 2000.
- [56] R. Huber, H. Ramoser, K. Mayer, H. Penz, and M. Rubik, "Classification of coins using an eigenspace approach," *Pattern Recognition Letters*, vol. 26, pp. 61-75, 2005.
- [57] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*: John Wiley & Son, 2001.
- [58] D. L. Donoho. (2000, Society, Los Angeles). *High-dimensional data analysis: The curses and blessings of dimensionality. Lecture delivered at the "Mathematical Challenges of the 21st Century" conference of The American Math.* Available: [http://www-stat.stanford.edu/~donoho/Lectures/AMS2000/MathChallengeSlides2\\*2.pdf](http://www-stat.stanford.edu/~donoho/Lectures/AMS2000/MathChallengeSlides2*2.pdf)
- [59] R. Bellman, *Adaptive Control Processes: A Guided Tour*: Princeton University Press, Princeton, 1961.
- [60] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*: Springer, 2001.

**References** .....

---

- [61] U.M. Fayyad and R. Uthurusamy, "Evolving data mining into solutions for insights," *Communications of the Association for Computing Machinery*, vol. 45, pp. 28 – 31, August 2002.
- [62] Ng. A. Y., "Preventing overfitting of crossvalidation data," *Proceedings of Fourteenth International Conference on Machine Learning*, 1997.
- [63] A. Jain and D. Zongker, "Feature Selection: Evaluation, Application, and Small Sample Performance," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, pp. 153-158, Feb. 1997.
- [64] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *J. Machine Learning Research*, vol. 3, pp. 1157-1182, 2003.
- [65] R. Kohavi and G.H. John, "Wrappers for Feature Subset Selection," *Artificial Intelligence*, vol. 97, pp. 273-324, 1997.
- [66] I. T. Jolliffe, *Principal Component Analysis*: Springer, 2002.
- [67] K. Fukunaga, *Introduction to Statistical Pattern Recognition*: Academic Press, 1990.
- [68] K. Etemad and R. Chellappa, "Face Recognition Using Discriminant Eigenvectors," *Proc. IEEE Int'l Conf. Acoustic, Speech, and Signal Processing*, vol. 4, pp. 2148-2151, 1996.
- [69] J.B. Tenenbaum, V. de Silva, J.C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.
- [70] M. Belkin, P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," *Proceedings Conference of Advances in Neural Information Processing Systems*, pp. 585–591, 2001.
- [71] Sam T. Roweis, K. Lawrence, Saul, "nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.
- [72] E. Kokiopoulou, Yousef Saad, "Orthogonal neighborhood preserving projections: a projection-based dimensionality reduction technique," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 2143–2156, 2007.
- [73] L. Sirovich, and M. Kirby, "Low-dimensional Procedure for the Characterization of Human Faces," *Journal of the Optical Society of America*, vol. 4, pp. 519-524, 1987.
- [74] Application of the Karhunen-Loeve Procedure, "Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 103-108, 1990.

**References** .....

---

- [75] B. Kepenekci, "Face Recognition Using Gabor Wavelet Transform," MSc. Thesis, METU, 2001.
- [76] B. Moghaddam, T. Jebara, and A. Pentland, "Bayesian face recognition," *Pattern Recognition*, vol. 33, pp. 1771–1782, 2000.
- [77] J. Ye, R. Janardan, Q. Li, "Two-dimensional linear discriminant analysis," *Advances in Neural Information Processing Systems 18 (NIPS)*, vol. 17, pp. 1569-1576, 2004.
- [78] X. He, P. Niyogi (2003). *Locality Preserving Projections (LPP)*. Available: <http://www.cs.uchicago.edu/research/publications/techreports/TR-2002-09>
- [79] L. Saul, S. Roweis, "Think globally, fit locally: unsupervised learning of nonlinear manifolds," *Journal of Machine Learning Research*, vol. 4, pp. 119–155, 2003.
- [80] J. Yang and J. Y. Yang, "From image vector to matrix: A straightforward image projection technique—imPCA vs. PCA," *Pattern Recognition*, vol. 35, pp. 1997–1999, 2002.
- [81] D. Tao, X. Li, X. Wu, W. Hu, S.J. Maybank, "Supervised tensor learning," *Knowledge and Information Systems*, vol. 13, pp. 1–42, 2007.
- [82] S. Z. Li, C. Zhao, X. Zhu, and Z. Lei, "3D+2D face recognition by fusion at both feature and decision levels," *Proc. IEEE Int. Workshop on Analysis and Modeling of Faces and Gestures*, pp. 43-53, 2005.
- [83] A. Colombo, C. Cusano, and R. Schettini, "3D face detection using curvature analysis," *Pattern Recognition*, vol. 39, pp. 444–455, Mar. 2006.
- [84] K. W. Bowyer, K. Chang, and P. Flynn, "A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition," *Computer Vision and Image Understanding*, vol. 101, pp. 1-15, 2006.
- [85] J. Ye, R. Janardan, Q. Li, "GPCA: an efficient dimension reduction scheme for image compression and retrieval," *The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 354–363, 2004.
- [86] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. Zhang, "Multilinear discriminant analysis for face recognition," *IEEE Transactions on Image Processing*, vol. 16 (1), pp. 212–220, 2007.
- [87] J. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, "Face recognition using kernel direct discriminant analysis algorithms," *IEEE Transactions on Neural Networks*, vol. 14 (1) pp. 117–126, 2003.

- [88] H. Lu, J. Wang, K.N. Plataniotis, *A review on face and gait recognition: system, data and algorithms*: CRC Press, Boca Raton, FloridaS. , 2009.
- [89] J. Li, L. Zhang, D. Tao, H. Sun, Q. Zhao, "A prior neurophysiologic knowledge free tensor-based scheme for single trial eeg classification," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 17 (2), pp. 107–115, 2009.
- [90] H. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, "Regularized common spatial patterns with generic learning for EEG signal classification," *Proceedings of 31st International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009.
- [91] J. Ye, T. Li, T. Xiong, R. Janardan, "Using uncorrelated discriminant analysis for tissue classification with gene expression data," *IEEE/ACM Transactions on Computational Biology Bioinformatics*, vol. 1 (4) pp. 181–190, 2004.
- [92] R.D. Green, L. Guan, "Quantifying and recognizing human movement patterns from monocular video images-part II: applications to biometrics," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14 (2), pp. 191–198, 2004.
- [93] D.R. Hardoon, J. Shawe-Taylor, "Decomposing the tensor kernel support vector machine for neuroscience data with structure labels," *Machine Learning*, vol. 79 (1–2), pp. 29–46, 2010.
- [94] H.S. Sahambi, K. Khorasani, "A neural-network appearance-based 3-D object recognition using independent component analysis," *IEEE Transactions on Neural Networks*, vol. 14 (1), pp. 138–149, 2003.
- [95] J. Ye, R. Janardan, and Q. Li, "Two dimensional linear discriminant analysis," *Advances in Neural Information Processing Systems (NIPS)*, pp. 1569–1576, 2004.
- [96] G. Shakhnarovich, B. Moghaddam, *Face recognition in subspaces*: S.Z. Li, A.K. Jain (Eds.), Handbook of Face Recognition, Springer-Verlag,, 2004.
- [97] D. Xu, S. Yan, L. Zhang, S. Lin, H.-J. Zhang, T.S. Huang, "Reconstruction and recognition of tensor-based objects with concurrent subspaces analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18 (1), pp. 36–47, 2008.
- [98] H. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, "A taxonomy of emerging multilinear discriminant analysis solutions for biometric signal recognition,"

- N.V. Boulgouris, K. Plataniotis, E. Micheli-Tzanakou (Eds.), *Biometrics: Theory, Methods, and Applications*, Wiley/IEEE, pp. 21–45, 2009.
- [99] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General Tensor Discriminant Analysis and Gabor Features for Gait Recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, pp. 1700-1715, Oct. 2007.
- [100] D. Tao, X. Li, X. Wu, S.J. Maybank, "Tensor rank one discriminant analysis a convergent method for discriminative multilinear subspace selection," *Neurocomputing*, vol. 71 (10–12), pp. 1866–1882, 2008.
- [101] H. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, "Uncorrelated multilinear principal component analysis for unsupervised multilinear subspace learning," *IEEE Transactions on Neural Networks*, vol. 20 (11), pp. 1820–1836, 2009.
- [102] H. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, "Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition," *IEEE Transactions on Neural Networks*, vol. 20 (1), pp. 103–123, 2009.
- [103] L.D. Lathauwer, B.D. Moor, J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal of Matrix Analysis and Applications*, vol. 21 (4), pp. 1253–1278, 2000.
- [104] L. Lathauwer, B. D. Moor, J. Vandewalle, "On the best rank-1 and rank  $(R_1, R_2, \dots, R_N)$  approximation of higher-order tensors," *Journal of Matrix Analysis and Applications*, vol. 21, pp. 1324-1342, 2000.
- [105] B.W. Bader, T.G. Kolda, "Algorithm 862: Matlab tensor classes for fast algorithm prototyping," *ACM Transactions on Mathematical Software*, vol. 32(4), pp. 635–653, 2006.
- [106] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal of Matrix Analysis and Applications*, vol. 21(4), pp. 1253-1278, 2000.
- [107] L. De Lathauwer and J. Vandewalle, "Dimensionality reduction in higher-order signal processing and rank- $(r_1, r_2, \dots, r_n)$  reduction in multilinear algebra," *Linear Algebra Appl., Special Iss. Linear Algebra Signal Image Process.*, vol. 391, pp. 31-55, 2004.
- [108] T. G. Kolda, "Orthogonal tensor decompositions," *SIAM Journal of Matrix Analysis and Applications*, vol. 23(1), p. 243{255, 2001.

*References* .....

---

- [109] E. Kofidis and P. A. Regalia, *Tensor approximation and signal processing applications* vol. I: Structured Matrices in Mathematics, Computer Science and Engineering, Contemp. Math. 280, V. Olshevsky, ed., AMS, Providence, RI, 2001.
- [110] L. De Lathauwer and J. Vandewalle, "Dimensionality reduction in higher-order signal processing and rank-( $R_1, R_2, \dots, R_N$ ) reduction in multilinear algebra," *Linear Algebra and its Applications*, vol. 391, pp. 31-55, 2004.
- [111] L. D. Lathauwer, "Signal Processing Based on Multilinear Algebra," PhD Thesis, Katholieke Universiteit Leuven, 1997.
- [112] H. Lu, "Multilinear subspace learning for face and gait recognition," Ph.D. Thesis, University of Toronto, 2008.
- [113] R. A. Harshman, "Foundations of the parafac procedure: Models and conditions for an "explanatory" multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1-84, 1970.
- [114] P. Kroonenberg and J. Leeuw, "Principal component analysis of three-mode data by means of alternating least squares algorithms," *Psychometrika*, vol. 45, pp. 69-97, 1980.
- [115] N. M. Faber, R. Bro, and P. K. Hopke, "Recent developments in CANDECOMP/PARAFAC algorithms: a critical review," *Chemometrics and Intelligent Laboratory Systems*, vol. 65(1), p. 119{137, 2003.
- [116] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Gait recognition through MPCA plus LDA," *Proc. Biometrics Symposium 2006*, pp. 1-6, 2006.
- [117] "The Olivetti & Oracle Research Laboratory Face Database of Faces. [Online]. Available: <http://www.cam-orl.co.uk/facedatabase.html>."
- [118] A.M. Martinez, R. Benavente, "The AR Face Database," *Technical Report CVC 24*, 1998.
- [119] I. Philips, H. Wechsler, J. Huang, and P. Rauss, "The FERET database and evaluation procedure for face recognition algorithms," *Image and Vision Computing*, vol. 16, pp. 295–306, 1998.
- [120] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence archive*, vol. 22, pp. 1090–1104, 2000.

- [121] C. Liu, "Capitalize on dimensionality increasing techniques for improving face recognition grand challenge performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28(5), p. 725-737, 2006.
- [122] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM systems Journal*, vol. 40, pp. 614-634, 2001.
- [123] A. Martin, T. K. G. Doddington, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," *Proceedings of EuroSpeech*, vol. 97(4), p. 1895-1898, 1997.
- [124] M.N.Do, M.Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," vol. 14(12), pp. 2091-2106, 2005.
- [125] M.N.Do, M.Vetterli, "The finite ridgelet transform for image representation," *IEEE Transactions on Image Processing*, vol. 12(1), pp. 16-28, 2003.
- [126] E.J.Candes,D.L.Donoho, "Curvelets: A surprisingly effective nonadaptive representation for objects with edges," *Curves and Surfaces*, vol. [online] Available on <http://www.Curvelet.org/papers/Curve99.pdf>, L. L. Schumaker et al. (eds), Vanderbilt University Press, Nashville, TN, , 2000.
- [127] K.C.Chung,S.C.Kee,S.R.Kim, "Face recognition using principal component analysis of Gabor filter responses," *Proceedings of the IEEE*, pp. 53-57, 1999.
- [128] C.F.Chen, Y.S.Tseng, C.Y.Chen, "Combination of PCA and wavelet transforms for face recognition on 2.5D images," *Proceedings of the Image and Vision Computing, NZ*, pp. 343-347, 2003.
- [129] G.Y.Tian,S.King,D.Taylor,S.Ward, "Wavelet based normalization for face recognition," *Proceedings of CGIM*, 2003.
- [130] G.C.Feng,P.C.Yuen,D.Q.Dai, "Human face recognition using PCA on wavelet subband," *Journal of Electronic Imaging*, vol. 9(2), pp. 226-233, 2000.
- [131] D. Q. Dai and H. Yan, "Wavelets and Face Recognition," *In Face Recognition*, K. Delac and M. Grgic ed., I-TECH Education and Publishing, Vienna, 2007.
- [132] J.T.Chien,C.C.Wu, "Discriminant wavelet faces and nearest feature classifiers for face recognition," *IEEE Transactionson Pattern Analysis and Machine Intelligence*, vol. 24(2), pp. 1644-1649, 2002.
- [133] V. PERLIBAKAS, "Face Recognition Using Principal Component Analysis and Wavelet Packet Decomposition," *informatica*, vol. 15(2), pp. 243 - 250, 2004.

**References** .....

---

- [134] Majid Safari, Mehrtash T. Harandi, Babak N. Araabi, "A SVM-based method for face recognition using a wavelet pca representation of faces," *Image Processing, ICIP apos; 04. International Conference,,* vol. 2 (24-27), pp. 853 – 856, October. 2004.
- [135] B.L.Zhang,H.Zhang,S.SamGe, "Face recognition by applying wavelet subband representation and kernel associative memory," *IEEE Transactions on Neural Networks*, vol. 15(1), pp. 166–177, 2004.
- [136] B. S. Hazim Kemal Ekenel, "Multiresolution face recognition," *Image and Vision Computing*, vol. 23, pp. 469–477, 2005.
- [137] C. Liu, D. Dai, H. Yan, "Local Discriminant Wavelet Packet Coordinates for Face Recognition," *Journal of Machine Learning Research*, pp. 1165-1195, 2007.
- [138] M. Zhao, P. Li, Z. Liu, "Face recognition based on wavelet transform weighted modular PCA," *Proceedings of the Congress in Image and Signal Processing*, pp. 589–593, 2008.
- [139] E.J.Candes, "What is curvelet," *Notices of American Mathematical Society*, vol. 50, pp. 1402–1403, 2003.
- [140] J. L. Starck, E. J. Candès and D. L. Donoho, "The Curvelet Transform for Image Denoising," *IEEE Transactions on Image Processing, ICIP* vol. 11 (6), pp. 670-684, June 2002.
- [141] E. J. Candès and D. L. Donoho, "Curvelets, multiresolution representation, and scaling laws," *SPIE Wavelet Applications in Signal and Image Processing VIII*, 2000.
- [142] D. L. Donoho, M. R. Duncan, "Digital curvelet transform: strategy, implementation and experiments," *Proceedings of SPIE*, vol. 4056, pp. 12-30, 2000.
- [143] F. Nencini, A. Garzelli, S. Baronti, et al, "Remote sensing image fusion using the curvelet transform," *Information Fusion*, vol. 8 (2), pp. 143-156, April 2007.
- [144] MChoi, R. Y. Kim, M. G. Kim, "The curvelet transform for image fusion," *Proceedings of ISPRS Congress, Istanbul*, 2004.
- [145] J. L. Starck, F. Murtagh, E. J.Candes,D. L. Donoho, "Gray and Colour Image Contrast Enhancement by the Curvelet Transform," *IEEE Transactions on Image Processing*, vol. 12(6), pp. 706-717, June 2003.

**References** .....

---

- [146] M. Manikandan, A. Saravanan, K. B. Bagan, "Curvelet transform based embedded lossy image compression," *Proceedings of ICSCN*, pp. 274–276, 2007.
- [147] A. Majumdar, "Image compression by sparse pca coding in curvelet domain," *Signal, Image and Video Processing*, vol. 3 (1), pp. 27-34, February 2009.
- [148] J.L.Starck, D.L.Donoho, E.J.Candes, "Very high quality image restoration by combining wavelets and curvelets," *Proceedings of SPIE*, vol. 4478, 2003.
- [149] S. Arivazhagan, L.Ganesan, T. G. Subash Kumar, "Texture classification using Curvelet Statistical and Co-occurrence Features," *Proceedings of the 18th International Conference on Pattern Recognition*, pp. 938-941, August 2006.
- [150] S. Arivazhagan, L.Ganesan, T. G. Subash Kumar, "Face recognition based on curvefaces," *Proceedings of Natural Computation*, 2007.
- [151] T.Mandal, A.Majumdar, Q.M.J.Wu, "Face recognition by curvelet based feature extraction," *roceedings of ICIAR*, vol. 4633, pp. 806–817, 2007.
- [152] A. Majumdar and A. Bhattacharya, "Face Recognition by Multiresolution Curvelet Transform on Bit Quantized Facial Images," *International Conference on Computational Intelligence and Multimedia Applications*, vol. 2 (13-15), pp. 209-213., 2008.
- [153] Tanaya Mandal, Q.M.J.Wu, "Face Recognition using Curvelet Based PCA," *ICPR*, 2008.
- [154] M. Rziza, M. El Aroussi, M. El Hassouni, S. Ghouzali and D. Aboutajdine, "Local Curvelet Based Classification Using Linear Discriminant Analysis for Face Recognition," *International Journal of Computer Science*, vol. 4 (1), pp. 72-77.
- [155] J. Xie, "Face Recognition Based on Curvelet Transform and LS-SVM," *Proceedings of the International Symposium on Information Processing (ISIP'09)*, pp. 140-143, Huangshan, P. R. China, 21-23 August 2009.
- [156] E.J.Candes, L. Demanet, D.L.Donoho, L.Ying, "Fast discrete curvelet transforms," *Multiscale Modeling and Simulations*, vol. 5(3), pp. 861–899, 2006.
- [157] E.J.Candes, F.Guo, "New multiscale transforms, minimum total variation synthesis: applications to edge-preserving image reconstruction," *Signal Processing: Applications to Edge*, vol. 82(11), pp. 1519–1543, 2002.
- [158] Q. Gao, L. Zhang and D. Zhang, "Directional independent component analysis with tensor representation," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

**References** .....

---

- [159] D. Zhang, Z. H. Zhou, and S. Chen, "Diagonal principal component analysis for face recognition," *Pattern Recognition*, vol. 39, pp. 140-142, 2006.
- [160] C. M. Bishop, *Neural Networks for Pattern Recognition: The Clarendon Press Oxford University Press*, New York, NY, USA, 1995.
- [161] S. S. Haykin, *Neural Networks: A Comprehensive foundation*: Prentice Hall, 1999.
- [162] M. T. Hagan, *Neural Network Design*: Brooks/Cole, 1996.
- [163] S. Samet and A. Miri, "Privacy-preserving back-propagation and extreme learning machine algorithms," *Data and Knowledge Engineering*, vol. 79-80, pp. 40–61, 2012.
- [164] G. B. Huang, Q. Y. Zhu and C. K. Siew, "Extreme Learning Machine: Theory and Applications," *Neurocomputing*, vol. 70, pp. 489-501, 2006.
- [165] G. B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 2 (2), pp. 107–122, 2011.
- [166] C.R.Rao and S.K.Mitra, *Generalized Inverse of Matrices and its Applications*: Wiley, New York, 1971.
- [167] Zhao GP, Shen ZQ, Man ZH, "Robust Input Weight Selection for Well-Conditioned Extreme Learning Machine," *International Journal of Information Technology*, vol. 17(1), pp. 1-13, 2011.
- [168] D. Serre, *Matrices, Theory and Applications*: Springer, New York, 2002.
- [169] "Matlab, The Language of Technical Computing, Version 7 (2004), <http://www.mathworks.com>."
- [170] M. Vasilescu and D. Terzopoulos, "Multilinear analysis for facial image recognition," *Proc. Int. Conf. on Pattern Recognition (ICPR'02)*, vol. 3, pp. 511–514, Aug. 2002.
- [171] D. X. S. Yan, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang, "Discriminant analysis with tensor representation," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. I, pp. 526-532, June 2005.
- [172] X. Han, Y. Chen, and X. Ruan, "Multilinear Tensor Supervised Neighborhood Embedding Analysis for View-Based Object Recognition," *PCM 2010, Part I*, vol. 6297, pp. 236-247, 2010.

**References** .....

---

- [173] X. Xu, X. Guan, D. Zhang, X. Zhang, W. Deng, and Z. Wang, "Feature fusion of palmprint and face via tensor analysis and curvelet transform," *OPTO-ELECTRONICS REVIEW*, vol. 20, pp. 138-147, 2012.
- [174] R. M. Mutelo, L. C. Khor, W. L. Woo, S. S. Dlay, "Two-dimensional reduction PCA: a novel approach for feature extraction, representation, and recognition," *Proceeding of SPIE in Visualization and Data Analysis*, vol. 6060, p. 60600F, 2006.