HUMAN VISUAL SYSTEM BASED OBJECT DETECTION AND RECOGNITION AND INTRODUCTION OF LOGARITHMIC LOCAL BINARY PATTERNS FOR FACE RECOGNITION

A thesis

Submitted by

Debashree Mandal

In partial fulfillment of the requirements

For the degree of

Master of Science

In Electrical Engineering

Tufts University

Date
August 2012

Adviser: Dr. Karen Panetta

ABSTRACT

This thesis aims at incorporating logarithmic image processing and the human visual response which is based on Weber's law into various image processing applications. Human Visual System (HVS) has been used in this thesis for image decomposition. Specifically HVS based image decomposition has been applied towards the development of a novel framework for object detection and recognition systems. Also Logarithmic Image Processing (LIP) has been used towards the development of novel feature vectors. Logarithmic Image Processing (LIP) replaces the linear arithmetic (addition, subtraction, and multiplication) with a non-linear one, which more accurately characterizes the nonlinearity of computer image arithmetic and is consistent with the Weber's Law and the saturation characteristics of the human visual system. Two systems have been developed. One of which detects eyes from facial images after performing morphological operations. It has been shown that extracting features from HVS decomposed images followed by a feature fusion results in a better rate of detection than when extracting features from the original image alone. This has also proved effective in images that have shadows near the eye region. This thesis also presents a novel approach to the problem of face recognition that combines the classical Local Binary Pattern (LBP) feature descriptors with image processing in the logarithmic domain and the human visual system. Particularly, we have introduced parameterized logarithmic image processing (PLIP) operators based LBP feature extractor. We have also used the human visual system based image decomposition to extract features from the decomposed images and combine those with the features extracted from the original images thereby enriching the feature vector set and obtaining improved rates of recognition. Experiments have clearly shown the superiority of the proposed scheme over classical LBP feature descriptors.

ACKNOWLEDGEMENT

I express my sincere thanks to Prof Karen Panetta, Prof Sos Agaian and Jamie Heller for their guidance and motivation throughout my graduate studies. I thank my friends at SIMLAB and the ECE Department for a wonderful time. Lastly a very special thanks to my father Ashoke Kumar Mandal, my mother Swapna Mandal and my husband Arindam Nath for being my source of inspiration.

TABLE OF CONTENTS

ABSTRACT ..... **II**

ACKNOWLEDGEMENT ..... **III**

LIST OF TABLES ..... **XIII**

INTRODUCTION **1**

1.1 MOTIVATION 1

1.2 OBJECTIVE 2

1.3 CONTRIBUTIONS 3

1.4 THESIS PLAN AND ORGANIZATION 5

REVIEW OF LOGARITHMIC IMAGE PROCESSING **8**

2.1 ROLE OF HUMAN VISION IN IMAGE PROCESSING 10

2.2 THE LOGARITHMIC IMAGE PROCESSING (LIP) MODEL 11

2.3 THE PARAMETERIZED LOGARITHMIC IMAGE PROCESSING (PLIP) MODEL 14

HUMAN VISUAL SYSTEM (HVS) BASED IMAGE DECOMPOSITION AND EDGE DETECTION **16**

3.1 REVIEW OF THE HUMAN VISUAL PHENOMENON 17

3.1.1 Image Decomposition based on HVS *23*

3.1.2 HVS based edge detection *26*

3.2 PARAMETER SELECTION FOR HVS 29

3.2.1 Alpha variation *29*

3.2.2 Beta variation *32*

REVIEW OF FEATURE EXTRACTION FROM IMAGES **37**

4.1 THE CELL EDGE DISTRIBUTION (CED) 39

4.2 THE PRINCIPAL PROJECTED EDGE DISTRIBUTION (PPED) 42

4.3 FEATURE FUSION USING CED AND PPED 45

4.4 HISTOGRAM OF ORIENTED GRADIENTS (HOG) 47

4.5 LOCAL BINARY PATTERNS (LBP) 50

4.6 FEATURE EXTRACTION BASED ON MORPHOLOGICAL IMAGE PROCESSING 53

4.7 FEATURE EXTRACTION BY PRINCIPAL COMPONENT ANALYSIS (PCA) 59

HUMAN VISUAL SYSTEM BASED OBJECT DETECTION AND RECOGNITION 64

5.1 GENERIC FRAMEWORK FOR HVS BASED OBJECT DETECTION AND RECOGNITION 65

HUMAN VISUAL SYSTEM BASED EYE DETECTION 68

6.1 FLOW DIAGRAM FOR HUMAN VISUAL SYSTEM BASED EYE DETECTION 69

6.1.1 Image Decomposition 70

6.1.2 Edge Detection 71

6.1.3 Morphological Operations 71

6.1.4 Fusion 72

6.1.5 Classification 72

6.2 RESULTS AND DISCUSSION 75

6.2.1 Tests using good images 75

6.2.2 Tests using Images having Non-Uniform Illumination and shadows around Eye regions 78

6.2.3 Tests on an Image after introducing Shadows and Tint 81

6.2.4 Tests on an Image where Subjects have Glasses on 82

6.2.5 Tests on Image having Multiple Faces 83

6.2.6 Tests on Image taken with Cell Phone Camera 84

6.2.7 Effect of the Size of the Structuring Element 84

6.3 EXECUTION TIME AND MEMORY USAGE 85

6.4 LIMITATIONS 86

6.5 SUMMARY 87

FACE RECOGNITION BASED ON LOGARITHMIC LOCAL BINARY PATTERNS 88

7.1 Face Recognition: Existing Methods 89
7.2 Schematic Diagram of a Facial Identification System 90
7.3 Local Binary Pattern (LBP) Feature Descriptors 91
7.3.1 Uniform Local Binary Patterns 92
7.3.2 Rotation Invariant Local Binary Patterns 93
7.4 Classifiers 94
7.5 Logarithmic-LBP and PLIP-LBP Based Face Recognition 96
7.5.1 Logarithmic LBP Feature Descriptors 97
7.5.2 PLIP-LBP Feature Descriptors 99
7.5.3 Logarithmic-LBP Feature Descriptors Based on HVS Image Decomposition 101
7.6 Simulation and Results using AT&T Laboratories face database 104
7.6.1 Using Uniform LBP feature descriptors and Chi-square distance statistic 104
7.6.2 Using Uniform LBP feature descriptors and Histogram Intersection statistic 110
7.6.3 Using Uniform LBP feature descriptors and weighted Chi-square statistic 111
7.6.4 Using Rotation Invariant LBP features with Weighted Chi-square statistic 113
7.6.5 Using Rotation Invariant LBP features with non-weighted Chi-square statistic 114
7.6.6 Using Rotation Invariant Uniform LBP with Chi-square distance statistic 115
7.7 Simulation and Results using the Yale face database 117
7.8 Examples of Recognition 118
7.9 Analysis and Discussion 121
7.10 Execution Time and Memory Usage 123
7.11 Summary 123

CONCLUSION 125

FUTURE WORK 127

APPENDIX A 129

REFERENCES 130

LIST OF FIGURES

FIGURE 3.1: THE INCREMENT THRESHOLD **ΔBT** AS A FUNCTION OF REFERENCE INTENSITY B.....19

FIGURE 3.2: LINEAR APPROXIMATION OF THE INCREMENT THRESHOLD **logΔBT** AS A FUNCTION OF REFERENCE INTENSITY **logB** 20

FIGURE 3.3: ORIGINAL GRAYSCALE IMAGE.....24

FIGURE 3.4: HVS BASED BACKGROUND IMAGES IN (A) WEBER REGION (B) DE VRIES-ROSE REGION (C) SATURATION REGION AND (D) DARK REGION.....24

FIGURE 3.5: HVS BASED GRADIENT IN (A) WEBER REGION (B) DE VRIES-ROSE REGION (C) SATURATION REGION AND (D) DARK REGION 25

FIGURE 3.6: (A) ORIGINAL IMAGE, HVS BASED IMAGES IN (B) WEBER REGION (C) DE VRIES-ROSE REGION (D) AND SATURATION REGION, (E) REMAINING IMAGE PIXELS (OTHER REGION) AND (F) RESULT OF ARITHMETIC ADDITION OF (B), (C), (D) AND (E).....25

FIGURE 3.6.1: ORIGINAL IMAGE 26

FIGURE 3.6.2: HVS BASED BACKGROUND IMAGES IN (A) WEBER REGION (B) DE VRIES-ROSE REGION (C) SATURATION REGION AND (D) DARK REGION.....27

FIGURE 3.6.3: HVS BASED GRADIENT IN (A) WEBER REGION (B) DE VRIES-ROSE REGION (C) SATURATION REGION AND (D) DARK REGION 27

FIGURE 3.6.4: EDGE MAP OBTAINED BY HVS.....28

FIGURE 3.7: (A) ORIGINAL IMAGES (B) RESULTS OF EDGE DETECTION BY HVS 28

FIGURE 3.8: HVS BASED BACKGROUND IMAGES IN (A) WEBER REGION (B) DE VRIES-ROSE REGION (C) SATURATION REGION AND (D) DARK REGION FOR $\boldsymbol{\alpha 1 = 0.1}$, $\boldsymbol{\alpha 2 = 0.3}$ AND $\boldsymbol{\alpha 1 = 0.9}$.....30

FIGURE 3.9: HVS BASED BACKGROUND IMAGES IN (A) WEBER REGION (B) DE VRIES-ROSE REGION (C) SATURATION REGION AND (D) DARK REGION FOR $\boldsymbol{\alpha 1 = 0.4}$, $\boldsymbol{\alpha 2 = 0.6}$ AND $\boldsymbol{\alpha 1 = 0.8}$ 30

FIGURE 3.10: EDGE MAP CORRESPONDING TO $\boldsymbol{\alpha 1 = 0.1}$, $\boldsymbol{\alpha 2 = 0.3}$ AND $\boldsymbol{\alpha 1 = 0.9}$ 31

FIGURE 3.11: EDGE MAP CORRESPONDING TO $\boldsymbol{\alpha 1 = 0.4}$, $\boldsymbol{\alpha 2 = 0.6}$ AND $\boldsymbol{\alpha 1 = 0.8}$.....31

FIGURE 3.12: EFFECT OF BETA VARIATION ON EDGE MAP.....33

FIGURE 3.14: HVS DECOMPOSED IMAGES IN (A) WEBER REGION (B) DE-VRIES ROSE REGION (C) SATURATION REGION AND (D) OTHER REGION FOR BETA = 0.002. 35
FIGURE 3.15: HVS DECOMPOSED IMAGES IN (A) WEBER REGION (B) DE-VRIES ROSE REGION (C) SATURATION REGION AND (D) OTHER REGION FOR BETA = 0.02. 35
FIGURE 3.15: HVS DECOMPOSED IMAGES IN (A) WEBER REGION (B) DE-VRIES ROSE REGION (C) SATURATION REGION AND (D) OTHER REGION FOR BETA = 0.2. 36
FIGURE 4.1: SCHEMATIC DIAGRAM OF THE CED FEATURE EXTRACTION ALGORITHM 40
FIGURE 4.2.1: ORIGINAL EXAMPLE IMAGE....41
FIGURE 4.2.2: CED FEATURE VECTOR EXTRACTED FROM IMAGE IN FIGURE 4.2.1. EDGE MAPS OBTAINED BY HVS EDGE DETECTION 41
FIGURE 4.3: SCHEMATIC DIAGRAM OF THE PPED FEATURE EXTRACTION ALGORITHM....44
FIGURE 4.4: PPED FEATURE VECTOR EXTRACTED FROM IMAGE IN FIGURE 4.2.1. EDGE MAPS OBTAINED BY HVS....44
FIGURE 4.5: SCHEMATIC DIAGRAM OF THE CED-PPED FEATURE EXTRACTION ALGORITHM 46
FIGURE 4.6: COMBINED CED-PPED FEATURE VECTOR EXTRACTED FROM IMAGE IN FIGURE 4.2.1. EDGE MAPS OBTAINED BY HVS 46
FIGURE 4.7: SCHEMATIC DIAGRAM OF THE HOG FEATURE EXTRACTION ALGORITHM 48
FIGURE 4.8: REPRESENTATION OF THE HOG FEATURE DESCRIPTOR ALGORITHM....49
FIGURE 4.9: (A) ORIGINAL IMAGE (B) GRADIENT MAGNITUDE (C) GRADIENT ANGLE (D) GRADIENT VECTOR AT EACH POINT RETAINS THE FEATURES OF THE FACIAL IMAGE (E) 36-ELEMENT HOG FEATURE VECTOR EXTRACTED FROM ONE R-HOG BLOCK. 50
FIGURE 4.10: BASIC LBP OPERATOR....51
FIGURE 4.11: CIRCULAR (8, 1), (16, 2) AND (8, 2) NEIGHBORHOOD 51
FIGURE 4.12: FACIAL IMAGE DIVIDED INTO 3X3 AND 6X6 BLOCKS 52
FIGURE 4.13: (A) GRAY SCALE IMAGE (B) LBP CODED IMAGE (C) $\boldsymbol{LBP8,2u2}$ FEATURE DESCRIPTOR FOR (A) 53
FIGURE 4.14: ORIGINAL BINARY IMAGE AND ITS DILATED VERSION 54
FIGURE 4.15: ORIGINAL BINARY IMAGE AND ITS ERODED VERSION 55
FIGURE 4.16: (A) ORIGINAL BINARY IMAGE (B) RESULT AFTER OPENING (C) RESULT AFTER CLOSING 55

FIGURE 4.17: SCHEMATIC DIAGRAM FOR BLOB EXTRACTION ALGORITHM....56

FIGURE 4.18.1: (A) ORIGINAL IMAGE (B) EDGE MAP BY HVS (C) IMAGE AFTER MORPHOLOGICAL PROCESSING (D) CLASSIFICATION OF BLOBS....56

FIGURE 4.18.2: (A) ORIGINAL IMAGE (B) EDGE MAP BY HVS (C) IMAGE AFTER MORPHOLOGICAL PROCESSING (D) CLASSIFICATION OF BLOBS....57

FIGURE 4.19: (A) ORIGINAL IMAGE (B) EDGE MAP USING THE CANNY EDGE DETECTOR (C) IMAGE AFTER MORPHOLOGICAL OPERATIONS....58

FIGURE 4.20: COMPARISON OF RESULTS USING THE CANNY EDGE DETECTOR WITH CUSTOM THRESHOLD VALUES (LOW THRESHOLD = .08; HIGH THRESHOLD = 0.2)....58

FIGURE 4.21: (A) ORIGINAL IMAGES (B) EDGE MAPS USING THE DEFAULT SOBEL EDGE DETECTOR (C) IMAGES AFTER MORPHOLOGICAL OPERATIONS....59

FIGURE 4.22: (A) ORIGINAL IMAGES (B) EDGE MAPS USING THE HVS EDGE DETECTOR (C) IMAGES AFTER MORPHOLOGICAL OPERATIONS....59

FIGURE 4.23: TRAINING IMAGES FROM AT&T LABORATORIES, CAMBRIDGE....62

FIGURE 4.24: MEAN FACIAL IMAGE....63

FIGURE 4.25: (A) GRAYSCALE IMAGE (B) MEAN SUBTRACTED IMAGE....63

FIGURE 4.24: TOP 20 EIGENFACES FROM THE DATABASE....63

FIGURE 5.1: SCHEMATIC DIAGRAM OF A TYPICAL OBJECT DETECTION SYSTEM....65

FIGURE 5.2: SCHEMATIC DIAGRAM OF OBJECT DETECTION SYSTEM INCORPORATING HVS BASED IMAGE DECOMPOSITION....66

FIGURE 6.1: SCHEMATIC DIAGRAM FOR EYE DETECTION BASED ON (A) HVS BASED IMAGE DECOMPOSITION AND MORPHOLOGICAL IMAGE PROCESSING (B) NON-HVS BASED PROCESSING....69

FIGURE 6.2: (A) ORIGINAL IMAGE AND ITS GRAYSCALE VERSION (B) RESULT AFTER DILATION AND EROSION IN THE WEBER REGION (C) RESULT AFTER DILATION AND EROSION IN THE "OTHER" REGION (D) RESULT AFTER DILATION AND EROSION IN THE SATURATION REGION, (E) IMAGE AFTER FUSION (LEFT) AND FINAL RESULT (RIGHT)....74

FIGURE 6.3: EXPERIMENTAL RESULTS USING WELL-ILLUMINATED IMAGES....76

FIGURE 6.4: VARIATION OF THE RATE OF CHANGE OF BLOB EXTRACTION RATE VS. THE HVS THRESHOLDING PARAMETER BETA: THE RESULTS OUTPERFORMS NON-HVS BASED PROCESSING FOR A RANGE OF BETA FROM .03-.09 77

FIGURE 6.5: (A) RESULTS AFTER MORPHOLOGICAL OPERATIONS IN THE WEBER REGION, (B) RESULTS AFTER MORPHOLOGICAL OPERATIONS IN THE DE-VRIES ROSE REGION, (C) RESULTS AFTER MORPHOLOGICAL OPERATIONS IN THE "OTHER" REGION, (D) FUSION OF (A), (B) AND (C), (D) FINAL RESULT (E) RESULTS AFTER MORPHOLOGICAL PROCESSING WITH NON-HVS METHOD, (F) FINAL RESULT WITHOUT HVS DECOMPOSITION ...79

FIGURE 6.6: EXPERIMENTAL RESULTS USING HVS BASED PROCESSING ON IMAGES HAVING NON-UNIFORM ILLUMINATION AND SHADOWS AROUND THE EYE REGIONS....80

FIGURE 6.7: EXPERIMENTAL RESULTS USING NON-HVS BASED PROCESSING ON IMAGES HAVING NON-UNIFORM ILLUMINATION AND SHADOWS AROUND THE EYE REGIONS....80

FIGURE 6.8: TEST ON IMAGE AFTER INTRODUCING SHADOWS AND TINT: HVS BASED PROCESSING DETECTS EYES SUCCESSFULLY 82

FIGURE 6.8: (A) ORIGINAL IMAGE (B) FEATURE EXTRACTION AFTER NON-HVS BASED PROCESSING (C) FEATURE EXTRACTION AFTER HVS BASED PROCESSING 83

FIGURE 6.9: (A) ORIGINAL IMAGE HAVING 2 FACES (B) RESULTS AFTER FEATURE EXTRACTION THROUGH HVS BASED PROCESSING (C) FINAL RESULT AFTER HVS BASED PROCESSING (D) FINAL RESULT AFTER NON-HVS BASED PROCESSING....83

FIGURE 6.10: (A) ORIGINAL IMAGE TAKEN WITH CELL PHONE CAMERA (B) FEATURE EXTRACTION USING NON-HVS BASED PROCESSING (C) FEATURE EXTRACTION BY HVS BASED PROCESSING (D) FINAL RESULT USING HVS BASED PROCESSING....84

FIGURE 6.11: (A) ORIGINAL IMAGE HAVING SIZE 78 X 78 (B) MORPHOLOGICAL OPERATIONS ON (A) WITH STRUCTURING ELEMENT HAVING SIZE 4 (C) MORPHOLOGICAL OPERATIONS ON (A) AFTER RESIZING IMAGE TO SIZE 450 X 350 (D) ORIGINAL IMAGE HAVING SIZE 2592 X 1936 (E) MORPHOLOGICAL OPERATIONS ON (D) WITH STRUCTURING ELEMENT HAVING SIZE 4 (F) MORPHOLOGICAL OPERATIONS ON (D) AFTER RESIZING IMAGE TO SIZE 450 X 350 85

FIGURE 7.1: SCHEMATIC DIAGRAM OF A FACIAL IDENTIFICATION SYSTEM 91

FIGURE 7.2: (A) ORIGINAL IMAGE AND LBP CODED IMAGES USING (B) CLASSICAL LBP OPERATOR, (C) LOGARITHMIC-LBP OPERATOR, (D) PLIP-LBP USING ORIGINAL IMAGE INTENSITIES, (E) PLIP-LBP AFTER LOGARITHMIC TRANSFORM AND (F) LOGARITHMIC LBP ON WEBER IMAGE 96

FIGURE 7.3A: (A) THE ORIGINAL IMAGE, THE LBP CODED IMAGE AND THE LBP FEATURE VECTOR EXTRACTED FROM THE ORIGINAL IMAGE; (B) IMAGE AFTER LOGARITHMIC TRANSFORM, THE LOGARITHMIC-LBP CODED IMAGE AND THE EXTRACTED LOGARITHMIC-LBP FEATURE VECTOR 98

FIGURE 7.3B: SCHEMATIC DIAGRAM: LOG-LBP BASED FACE RECOGNITION.....98

FIGURE 7.4A: (A) PLIP- LBP CODED IMAGE AND THE PLIP-LBP FEATURE VECTOR EXTRACTED FROM THE ORIGINAL IMAGE; (B) PLIP-LBP CODED IMAGE AFTER LOGARITHMIC TRANSFORM AND THE EXTRACTED FEATURE VECTOR 100

FIGURE 7.4B: SCHEMATIC DIAGRAM: PLIP-LBP BASED FACE RECOGNITION.....100

FIGURE 7.5A: (A) THE WEBER IMAGE, THE LBP CODED WEBER IMAGE AND THE LBP FEATURE VECTOR EXTRACTED FROM THE WEBER IMAGE; (B) WEBER IMAGE AFTER LOGARITHMIC TRANSFORM, THE LBP CODED WEBER IMAGE AFTER LOGARITHMIC TRANSFORM AND THE EXTRACTED LBP FEATURE VECTOR 102

FIGURE 7.5B: SCHEMATIC DIAGRAM: HVS-LBP BASED FACE RECOGNITION 102

FIGURE 7.6: (A) THE CLASSICAL LBP OPERATOR (B) LOGARITHMIC LBP OPERATOR (C) PLIP-LBP OPERATOR USING THE ORIGINAL IMAGE INTENSITIES (D) PLIP-LBP OPERATORS AFTER APPLYING LOGARITHMIC TRANSFORM (E) LOGARITHMIC LBP EXTRACTED FROM THE WEBER IMAGE 103

FIGURE 7.7: COMPARISON OF AVERAGE RECOGNITION RATE TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING UNIFORM LBP WITH CHI-SQUARE STATISTIC 109

FIGURE 7.8: COMPARISON OF AVERAGE RECOGNITION RATE TAKING RANDOMLY SELECTED 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING UNIFORM LBP WITH CHI-SQUARE STATISTIC 109

FIGURE 7.9: COMPARISON OF AVERAGE RECOGNITION RATE TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING UNIFORM LBP WITH HISTOGRAM INTERSECTION STATISTIC.....111

FIGURE 7.10: COMPARISON OF AVERAGE RECOGNITION RATE TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING UNIFORM LBP WITH WEIGHTED CHI-SQUARE STATISTIC.....113

FIGURE 7.11: COMPARISON OF AVERAGE RECOGNITION RATE TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING ROTATION INVARIANT UNIFORM LBP WITH CHI-SQUARE STATISTIC 116

FIGURE 7.12: COMPARISON OF AVERAGE RECOGNITION RATE TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND NEST 5 IMAGES FOR TESTING. USING UNIFORM LBP WITH CHI-SQUARE STATISTIC 118

FIGURE 7.13: COMPARISON OF FACE RECOGNITION USING CLASSICAL AND LOGARITHMIC LBP USING 5 IMAGES FOR TRAINING FROM AT&T FACE DATABASE 119

FIGURE 7.14: COMPARISON OF FACE RECOGNITION USING CLASSICAL AND PLIP-LBP USING 3 IMAGES FOR TRAINING FROM AT&T FACE DATABASE 119

FIGURE 7.15: COMPARISON OF FACE RECOGNITION USING CLASSICAL AND HVS-LBP USING 1 IMAGE FOR TRAINING FROM AT&T FACE DATABASE 120

FIGURE 7.16: COMPARISON OF FACE RECOGNITION USING CLASSICAL AND HVS-LBP USING 4 IMAGES FOR TRAINING FROM YALE FACE DATABASE 120

LIST OF TABLES

TABLE 3.1: SUMMARY OF BETA VALUES.....36

TABLE 6.1: COMPARISON OF DETECTION RATE IN FEATURE EXTRACTION STEP 76

TABLE 6.2: COMPARISON OF DETECTION RATE IN THE FINAL EYE DETECTION STEP..... 77

TABLE 6.3: COMPARISON OF DETECTION RATE IN THE FINAL EYE DETECTION STEP USING IMAGES HAVING NON-UNIFORM ILLUMINATION AND SHADOWS AROUND THE EYE REGIONS..... 81

TABLE 6.4: COMPARISON OF THE EXECUTION TIME FOR HVS VS. NON-HVS APPROACH 85

TABLE 7.1: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS LOGARITHMIC LBP TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 1 AND CHI-SQUARE DISTANCE STATISTIC 105

TABLE 7.2: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS FUSED LOGARITHMIC LBP TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 1 AND CHI-SQUARE DISTANCE STATISTIC..... 106

TABLE 7.3: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. LOGARITHMIC LBP TAKING RANDOMLY CHOSEN 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 1 AND CHI-SQUARE DISTANCE STATISTIC..... 106

TABLE 7.4: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS LOGARITHMIC LBP ON WEBER IMAGE TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING

UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 1 AND CHI-SQUARE DISTANCE STATISTIC 107

TABLE 7.5: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. FUSED LOG-LBP (LOG-LBP FEATURE VECTOR FROM THE ORIGINAL IMAGE IS FUSED WITH THE LOG-LBP FEATURE VECTOR OBTAINED FROM THE WEBER IMAGE).TEST CONFIGURATION #1 TAKES FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND LAST 5 IMAGES FOR TESTING. TEST CONFIGURATION #2 TAKES RANDOMLY SELECTED 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 1 AND CHI-SQUARE DISTANCE STATISTIC 108

TABLE 7.6: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS PLIP-LBP TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 1 AND CHI-SQUARE DISTANCE STATISTIC 108

TABLE 7.7: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. PLIP-LBP TAKING RANDOMLY SELECTED 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 1 AND CHI-SQUARE DISTANCE STATISTIC..... 108

TABLE 7.8: COMPARISON OF THE RATES OF RECOGNITION USING THE HISTOGRAM INTERSECTION DISTANCE STATISTIC USING DIFFERENT TYPES OF LBP OPERATORS. FIRST 5 IMAGES ARE USED FOR TRAINING AND THE REMAINING 5 IMAGES FOR TESTING. USING UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 1 110

TABLE 7.9: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. LOGARITHMIC LBP TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 1 AND WEIGHTED CHI-SQUARE DISTANCE STATISTIC 111

TABLE 7.10: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. FUSED LOGARITHMIC LBP TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 1 AND CHI-SQUARE DISTANCE STATISTIC..... 112

TABLE 7.11: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. FUSED LOG-LBP (LOG-LBP FEATURE VECTOR FROM THE ORIGINAL IMAGE IS FUSED WITH THE LOG-LBP FEATURE VECTOR OBTAINED FROM THE WEBER IMAGE) TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND LAST 5 IMAGES FOR TESTING. USING UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 1 AND WEIGHTED CHI-SQUARE DISTANCE STATISTIC 112

TABLE 7.12: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. PLIP-LBP TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 1 AND WEIGHTED CHI-SQUARE DISTANCE STATISTIC..... 112

TABLE 7.13: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. LOGARITHMIC LBP TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING ROTATION INVARIANT LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 2 AND WEIGHTED CHI-SQUARE DISTANCE STATISTIC 113

TABLE 7.14: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. FUSED LOGARITHMIC LBP TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING ROTATION INVARIANT LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 2 AND WEIGHTED CHI-SQUARE DISTANCE STATISTIC 114

TABLE 7.15: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. PLIP-LBP TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING ROTATION INVARIANT LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 2 AND CHI-SQUARE DISTANCE STATISTIC 114

TABLE 7.16: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. FUSED LOG-LBP (LOG-LBP FEATURE VECTOR FROM THE ORIGINAL IMAGE IS FUSED WITH THE LOG-LBP FEATURE VECTOR OBTAINED FROM THE WEBER IMAGE) TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND LAST 5 IMAGES FOR TESTING. USING ROTATION INVARIANT LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 2 AND CHI-SQUARE DISTANCE STATISTIC 115

TABLE 7.17: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. LOGARITHMIC LBP TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING ROTATION INVARIANT UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 2 AND CHI-SQUARE DISTANCE STATISTIC 115

TABLE 7.18: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. FUSED LOGARITHMIC LBP TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING ROTATION INVARIANT UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 2 AND CHI-SQUARE DISTANCE STATISTIC 115

TABLE 7.19: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. PLIP-LBP TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND REMAINING 5 IMAGES FOR TESTING. USING ROTATION INVARIANT UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 2 AND CHI-SQUARE DISTANCE STATISTIC..... 116

TABLE 7.20: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. FUSED LOG-LBP (LOG-LBP FEATURE VECTOR FROM THE ORIGINAL IMAGE IS FUSED WITH THE LOG-LBP FEATURE VECTOR OBTAINED FROM THE WEBER IMAGE) TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND LAST 5 IMAGES FOR TESTING. USING ROTATION INVARIANT UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 2 AND CHI-SQUARE DISTANCE STATISTIC 116

TABLE 7.21: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. LOGARITHMIC LBP TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND NEXT 5 IMAGES FOR TESTING. USING UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 1 AND CHI-SQUARE DISTANCE STATISTIC 117

TABLE 7.22: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. FUSED LOG-LBP (LOG-LBP FEATURE VECTOR FROM THE ORIGINAL IMAGE IS FUSED WITH THE LBP FEATURE VECTOR OBTAINED FROM THE WEBER IMAGE) TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND NEXT 5 IMAGES FOR TESTING. USING UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 1 AND CHI-SQUARE DISTANCE STATISTIC 117

TABLE 7.23: COMPARISON OF THE RATE OF RECOGNITION OF CLASSICAL LBP VS. PLIP-LBP TAKING FIRST 5 IMAGES PER SUBJECT FOR TRAINING AND NEXT 5 IMAGES FOR TESTING. USING UNIFORM LBP TAKING 8 POINTS ON A CIRCULAR NEIGHBORHOOD OF RADIUS 1 AND CHI-SQUARE DISTANCE 118

TABLE 7.24: COMPARISON OF THE RATE OF RECOGNITION FOR THE MODIFIED LBP OPERATORS USING CHI-SQUARE DISTANCE STATISTIC 121

TABLE 7.25: COMPARISON OF THE RATE OF RECOGNITION FOR THE MODIFIED LBP OPERATORS ACROSS DIFFERENT CLASSIFIERS 122

TABLE 7.26: COMPARISON OF THE RATE OF RECOGNITION FOR THE MODIFIED LBP OPERATORS ACROSS DIFFERENT CLASSIFIERS 123

CHAPTER 1

INTRODUCTION

1.1 Motivation

In the past few decades there has been a tremendous growth in the field of image processing leading to the development of real time computer vision systems. Computer vision essentially aims at duplicating the effect of human vision by perceiving and understanding an image [1]. A human observer perceives an object by distinguishing it from its immediate background. One of the primary objectives of a computer vision system is to effectively mimic the visual response of the human eye. A human observer can easily detect an object from an image or video even under partial occlusion. For example if we consider the problem of facial recognition, we'll see that the task of identifying a person is not that difficult for a human. However the same task becomes extremely challenging for a computer vision system due to the absence of a unique representation of the object, which in this case is the face. Object detection and recognition algorithms are often applied for a myriad of practical purposes like crowd surveillance, entrance security, video context indexing, automatic detection of threat objects in airport security systems, finger print identification, detecting eyes to determine driver fatigue while driving a vehicle and many more. However the state of the art object detection and recognition algorithms do not take into account the non-linearity of the human visual response. The way in which a human observer perceives an object is extremely complex and largely depends on the background on which the object resides. From our day to day experience we know that in a very dark background, the slightest ray of light would cause a visual sensation whereas in a well-illuminated surrounding, that same small change might not cause a visual sensation. Moreover the visual response of the

eye is non-linear and hence an accurate simulation of the human visual system in image processing applications is an extremely challenging task. The logarithmic image processing (LIP) framework introduced in [2] replaces the linear arithmetic operations with a non-linear one which more accurately characterizes the response of the human eye. Inspired by the success of the LIP model towards various image processing applications like image enhancement, image restoration, image denoising and a simple but powerful texture descriptor, called Local Binary Pattern (LBP), we have proposed a system that does face recognition using novel modified logarithmic LBP operators. Also a mathematical formulation of the human visual system enables to decompose images. We have used this concept towards the development of a generic framework for object detection and recognition and have developed an eye detection system based on this framework.

1.2 Objective

The objective of this thesis is to apply logarithmic image processing and human visual system based image processing algorithms towards the development of object detection and recognition systems. This includes an in-depth study of the existing feature extraction algorithms to understand their advantages and disadvantages. A major part of the thesis has been devoted in understanding the role of human visual response in various image processing applications like edge detection, image decomposition, feature extraction, object detection and recognition. As mentioned in the previous section, the response of the human eye is extremely complex and depends largely on the background against which the object is visualized. Hence non-uniform illumination, shadows etc. present in image tend to affect the efficiency of image processing algorithms. Thus the aim was to apply logarithmic and HVS based processing and compare

results with classical approaches. A major objective was to develop novel feature vectors that use the non-linear LIP framework for face recognition. This framework is consistent with the non-linear characteristics of the human visual system and hence our objective was to improve the robustness of facial recognition by utilizing the LIP framework.

In summary the objectives are:

- Applying human visual system based image processing towards edge detection, image decomposition, feature vectors extraction, development of object detection and recognition systems.
- Analyzing the effect of different parameters for human visual system based image processing and parameter tuning for improved performance.
- Development of human visual system based object detection and recognition systems and comparing their performance with traditional systems that do not take into account human visual system based image processing.
- Development of novel feature vector that combines the classical Local Binary Pattern (LBP) feature descriptors with image processing in the logarithmic domain and the human visual system.

1.3 Contributions

The contributions are:

- Introduction of human visual system (HVS) based image decomposition towards feature extraction from images.

- Introduction of a new framework involving human visual response based image decomposition for object detection and recognition systems.
- An analysis of parameter selection in human visual system based image decomposition.
- A literature review of the state of the art feature extraction algorithms from images.
- Development of an eye detection system using human visual system based image decomposition and morphological image processing. The system introduces human visual system based image decomposition into an existing system and compares the performance with the original non-HVS based system. The system is tested using images having and non-uniform illumination and shadows near the eye region. It is shown that by suitably selecting parameters for HVS decomposition, the performance of the system surpasses that of the original system when testing images having non-uniform illumination.
- Introduction of a novel approach to the problem of face recognition that combines the classical Local Binary Pattern (LBP) feature descriptors with image processing in the logarithmic domain. Novel logarithmic-LBP feature descriptors have been introduced
- Introduction of parameterized logarithmic image processing (PLIP) operators based LBP feature extractor.
- Introduction of human visual system based LBP operator, which is based on the Weber's law. In this approach features extracted from the decomposed images are combined with the features extracted from the original images thereby enriching the feature vector set and obtaining improved rates of recognition.

1.4 Thesis plan and organization

We start the thesis by presenting an overview of the non-linear LIP framework for image processing. Next we present an overview of the mathematical framework for human visual system based edge detection and image decomposition. The state of the art feature vector extraction algorithms are also studied. We then propose a novel framework for object detection and recognition that can be used to enrich the feature vector set by applying HVS based image decomposition. We essentially developed two systems that detect objects from images based on human visual response. One of them detects eyes from facial images and we show the effectiveness of the system on images having non-uniform illumination. The second system does face recognition based on the novel logarithmic LBP based feature descriptors that have been introduced as a part of this thesis.

Thesis Plan and Methodology

- Review of the non-linear logarithmic framework for image processing.
- Understanding the pros and cons of the conventional edge detection algorithms and understanding the motivation for using human visual system based image processing for edge detection.
- Analyzing the role of parameter selection for HVS based image decomposition.
- A literature review of the existing feature extraction algorithms from images.
- Introduction of HVS based edge detection for edge based feature extraction from images.
- Introduction of HVS based image decomposition for feature extraction from images.
- Introduction of novel framework for human visual system based object detection and recognition

- Introduction of HVS based image decomposition for the development of eye detection system. Testing with images having shadows non-uniform illumination around the eyes region.
- Novel approach to face recognition that combines the classical Local Binary Pattern (LBP) feature descriptors with image processing in the logarithmic domain and the human visual system.
- Comparison of performance of facial recognition using modified LBP based feature extractors and classical LBP feature extractors using publicly available face databases.

The remaining part of the thesis is organized as follows: Chapter 2 gives an overview of the logarithmic image processing framework. We will show how this framework is consistent with the non-linear and saturation characteristics of the human visual system.

Chapter 3 gives an overview of the mathematical framework for human visual system based image decomposition. The effect of parameter selection on the algorithm is also presented.

Chapter 4 presents a review of the state of the art feature extraction algorithms of images. A feature vector is essentially an n-dimensional vector of numerical features that represent some object in an image. Many algorithms in machine learning require a numerical representation of objects, since such representations facilitate processing and statistical analysis.

Chapter 5 presents the novel framework for human visual system based object detection and recognition from images.

In chapter 6 we present a novel eye detection system based on human visual system and morphological image processing. We show how incorporating human visual system based image decomposition improves the results of the original algorithm. We present results of our testing

and compare with the original system that does not involve human visual system based processing for eye detection. We also show that the algorithm is effective on images having non-uniform illumination and shadows around the eye region.

In chapter 7 we introduce a novel approach towards facial recognition that combines the classical Local Binary Pattern (LBP) feature descriptors with image processing in the logarithmic domain and the human visual system. We compare performance of face recognition system using the modified LBP operators and the classical LBP operators. We perform our experiments using publicly available face databases.

CHAPTER 2

REVIEW OF LOGARITHMIC IMAGE PROCESSING

Image processing involves the transformation of an image from one form into another. The result may be a new image or may take the form of an abstraction, parameterization, or a decision [3]. Image processing includes many applications, such as image enhancement, edge detection, object recognition, and noise reduction. The most important aspect of an image processing technique is that the image processing framework must be physically consistent with the nature of the images and the mathematical rules and structures must be compatible with the information to be processed [4]. Jain [5] has clearly shown the interest and power of mathematics for image representation and processing. Granrath [6] has recognized that the human visual laws and models play an important role in image processing. Marr [7] has pointed out that, to develop an effective computer vision technique, the following three points must be considered: (1) what are the particular operations to be used and why? (2) How the images can be represented? And (3) what implementation structure can be used? Moreover, Schreiber [8] has argued that image processing is an application field and not a fundamental science. Thus an image processing framework must satisfy the following four main fundamental requirements [9]

1. It is based on a physically (and/or psychophysically) relevant image formation model
2. Its mathematical structures and operations are both powerful and consistent with the physical nature of the images, that is, with the image formation and combination laws
3. Its operations are computationally effective
4. It is practically fruitful in the sense that it enables to develop successful applications in real situations

Traditionally, image processing makes use of linear operations to manipulate images, but since computer arithmetic (computer addition, multiplication and others) is inherently a non-linear process, accuracy issues can arise [10]. As an example we can consider the addition of two images. In the digital world the addition of two images may produce "out-of-range" problems, because a value above the saturation threshold (for example, for 8 bits, the maximum value would be 255) is likely to be obtained when two images are digitally added and is likely to be clipped causing a loss of information. Saturation of these values to the maximum often happens in digital image processing, but those extreme values are actually never reached in natural images, since our retina, which acts as a natural sensor, works in a logarithmic mode following the Fechner's Law [11]. Moreover linear operations typically do not yield results consistent with the physical phenomena [10].

Thus image processing specific arithmetic operations were introduced to address the loss of information issue [12]. One attempt at resolving computer arithmetic issues was proposed by Ritter et al. The image algebra was designed to provide a mathematical framework to support implementation, comparison, and analysis of image processing transformations [13] [14]. It consists of a core set of 14 operations [15]. This image algebra has the standard point-wise addition, multiplication, and it focuses on developing operations to implement convolutions. Jourlin and Pinoli introduced the Logarithmic Image Processing (LIP) model [2] . The LIP model replaces the linear arithmetic (addition, subtraction, and multiplication) with a non-linear one, which more accurately characterizes the nonlinearity of computer image arithmetic. Moreover LIP model is consistent with the Weber's Law and the saturation characteristics of the human visual system [16].

2.1 Role of Human Vision in Image Processing

The human visual perception plays an important role in image processing applications [6] [7]. However there exists a large gap between the strong ability of human vision to perform difficult perceptual tasks, such as pattern recognition or image correlation and the weakness of even sophisticated algorithms to successfully address such problems [17]. One of the major reason for this is that the human vision system is complex and still remains partially unknown or not known enough [17]. In this section we'll present a review of the two laws that formulate the human visual perception towards light intensity.

Weber Law: Psychophysicist Weber [18] [19] established that the response to light intensity of the human visual system is non-linear. He argued that the human visual detection depends on the ratio, rather than the difference, between the light intensity values F and $F + \Delta F$, where ΔF is the so-called "just noticeable difference," which is the amount of light necessary to add to a visual test field of intensity value F such that it can be discriminated from a reference field of intensity value F. Weber's law can be expressed as

$$\frac{\Delta F}{F} = W \tag{2.1}$$

where W is the Weber's constant.

Fechner's Law: Fechner explained [11] the non-linearity of human visual sensation as follows: in order to produce incremental arithmetic steps in sensation, the light intensity must grow geometrically. He proposed the following relationship between the light intensity F (stimulus) and the brightness B (sensation):

$$\Delta B = k \frac{\Delta F}{F} \tag{2.2}$$

where ΔF is the increment light that produces increment ΔB of sensation and k is a constant. The Fechner's law can then be expressed as:

$$B = k' \frac{F}{F_{min}} \tag{2.3}$$

where k' is an arbitrary constant and F_{min} is the absolute threshold of human eye [96].

2.2 The Logarithmic Image Processing (LIP) Model

The logarithmic image processing (LIP) model is a mathematical framework based on abstract linear mathematics which provides a set of specific algebraic and functional operations that can be applied to the processing of intensity images valued in a bounded range [20]. The LIP model has proved to be consistent with the laws of the human visual system and has been successfully applied to image processing applications like image enhancement, image restoration, three-dimensional image reconstruction, edge detection and image segmentation.

In the LIP model, the brightness of an image is regarded as the intensity of light that passes through a light filter with absorption function $f(i,j)$ [21]. An absorption function is defined as the percentage of incident light being absorbed by the light filter. Jourlin and Pinoli called this function the gray tone function. The value of the gray tone function at spatial location (i,j) is thus called the gray tone. Thus an image can also be represented by the absorption function which describes the opacity of the light filter. Since the brightness of an image is usually restricted to an interval (typically [0,255] for an 8-bit image), then the gray tone function is also restricted in the real interval [0, M). Here, the value zero means that there is no absorption, while the value M corresponds to a totally opaque image. For an 8-bit image, M is 256. Therefore, the

gray tone is also in that interval. Using the gray tone function, the addition of two images can be performed by putting two light filters together. The gray tone function can thus be expressed as:

$$g(i,j) = M - f(i,j) \tag{2.4}$$

Where f(i, j) is the original image function, g(i, j) is the output gray tone function, and M is the maximum value of the range. The addition of two gray tones a and b, and the scalar multiplication of a by a positive real number α are defined in terms of the usual real function operations as:

$$a \oplus b = a + b - \frac{ab}{M} \tag{2.5}$$

$$\alpha \otimes a = M - M(1 - {}^{a}/_{M})^{\alpha} \tag{2.6}$$

If the gray tone functions are defined on (-∞,M), then subtraction of gray tones are given by

$$a \ominus b = M \frac{a-b}{M-b} \tag{2.7}$$

Using the subtraction operation, Jourlin and Pinoli [22] have proposed a definition of the contrast between two neighboring pixels:

$$c(f,g) = \max(f,g) \ominus \min(f,g) \tag{2.8}$$

This definition of contrast is consistent with the Weber's law. Jourlin and Pinoli have proved that the set of gray tones defined on (-∞,M) with operations $\oplus$ and $\otimes$ is a real vector space over the real number field. As a result, the set of gray tone is closed under the above operations. This ensures that the addition of two images, or the scalar multiplication of an image by a positive real number, results in a new image which is within the interval [0, M). This is very desirable for

many image processing problems such as image enhancement, image restoration and image data compression. Pinoli [2] has also proved that the real vector space is algebraically and topologically isomorphic to the real number space by the isomorphic transform given by:

$$\varphi(a) = -M.ln(1 - \frac{a}{M}) \tag{2.9}$$

The inverse isomorphic transform is given by,

$$\phi^{-1}(a) = M.\left[1 - \exp(\frac{-a}{M})\right] \tag{2.10}$$

The isomorphic transform serves as a powerful tool for developing the LIP model and for simplifying the analysis and the implementation of the LIP model based image processing algorithms. For example the multiplication of two gray tones is given by:

$$a \otimes b = \varphi^{-1}(\varphi(a).\varphi(b)) \tag{2.11}$$

The LIP model has been successfully used for different types of image processing algorithms like image enhancement, edge detection, and image restoration. In [22], LIP methods are used to enhance cervical smears for detection of cancerous cells, performing contrast enhancement and edge detection. In [2] and [23], LIP has been used for various image processing applications like noise removal, range control, and other methods for restoration and enhancement. LIP model has also been extended for color images and is used for the enhancement of transmitted medical images [24].

2.3 The Parameterized Logarithmic Image Processing (PLIP) Model

The LIP model however has some limitations as discussed in [10]. When two visually pleasing images are added together, the output image may not be visually pleasing. When linear arithmetic is used, added images are always brighter than the originals, which can result in images that are overall too bright. When classical LIP arithmetic is used, added images are always darker than the originals, which can result in images that are overall too dark. Ideally, added images will be representative of the originals in terms of overall brightness without unnaturally becoming too dark or too bright. Panetta et al demonstrated this problem in [10].

The parameterization of the LIP model was introduced by Panetta et al [10]. Each of the three operations addition, subtraction and multiplication are parameterized separately in this model. Measures of image enhancement [25] are utilized to judge performance along with mean-squared error measurements to determine the best parameters. This ensures that the PLIP operations should not visually damage an image. The PLIP model can be summarized as:

$$a \oplus b = a + b - \frac{ab}{\gamma(M)} \tag{2.12}$$

$$a \,\Theta\, b = k(M)\frac{a-b}{k(M)-b} \tag{2.13}$$

$$a \otimes b = \varphi^{-1}(\varphi(a).\varphi(b)) \tag{2.14}$$

$$\varphi(a) = -\lambda(M).ln^{\beta}(1 - \frac{a}{\lambda(M)}) \tag{2.15}$$

$$\phi^{-1}(a) = \lambda(M).\left[1 - \exp\left(\frac{-a}{\lambda(M)}\right)^{1/\beta}\right] \tag{2.16}$$

Where $\oplus$ is used as PLIP addition, Θ is used for PLIP subtraction and $\otimes$ is used for PLIP multiplication. Also, a and b are the grey tone pixel values and M is the maximum value of the

range, and β is a constant. $\gamma(M)$, $k(M)$, and $\lambda(M)$ are all arbitrary functions. We use the linear case, such that they are functions of the type $\gamma(M) = AM + B$, where A and B are integers, however any arbitrary function will work. It is to be noted that the PLIP equations revert to the classical case when $A = 1,\ \beta = 1\ and\ B = 0$. Also the PLIP arithmetic satisfies the fundamental requirements of an image processing framework and introduces a fifth constraint; the framework must not damage either image; i.e. when a "good" image is added to a "good" image, the output must be a "good" image. The parameters for PLIP can be selected using image enhancement measures [10] and the best values of A and B were experimentally determined to be any combination such that $\gamma(M), k(M), and\ \lambda(M) = 1026$ and the best value of β was determined to be $\beta = 2$.

PLIP operators were successfully utilized for image decomposition, image enhancement and edge detection. In this thesis PLIP operators have been extended to extract features from images. Specifically we have integrated PLIP operators with feature extraction by local binary patterns (LBP) and applied them towards face recognition.

CHAPTER 3

HUMAN VISUAL SYSTEM (HVS) BASED IMAGE DECOMPOSITION AND EDGE DETECTION

The way in which a human observer sees an object largely depend on the background on which the object is located. One of the very first image processing applications which incorporated this concept was edge detection. The classical edge detection algorithms like Sobel [26] [27], Roberts [28], Prewitt [29], Canny [30] [31] in general have two broad steps: gradient image calculation and thresholding. Most of the classical methods with the exception of canny choose a single global threshold for the entire image to detect the final edge map from the gradient image. If the value of the threshold is too high, the final edge map may have less useful information. On the other hand choosing a very low threshold may result in false edges in the edge map. The canny algorithm tries to solve the problem by performing an adaptive hysteresis thresholding with the choice of two thresholds. However the thresholds values are once again not dynamic and remain constant for the entire image. This poses a difficulty in edge detection on images that have varying illumination or shadows in them. Threshold selection is a very important step in image processing algorithms concerning segmentation, edge detection and image decomposition. A variety of techniques have been proposed in this regard [32] [33]. In an ideal case, the histogram has a deep and sharp valley between two peaks representing objects and background, respectively, so that the threshold can be chosen at the bottom of this valley. However, for most real pictures, it is often difficult to detect the valley bottom precisely, especially in such cases as when the valley is flat and broad, imbued with noise, or when the two peaks are extremely unequal in height, often producing no traceable valley. The well-known Otsu method for image segmentation chooses a threshold based on the image histogram so as to maximize the

separability of the resultant classes in gray levels. The choice of an appropriate threshold plays a very important role in edge detection algorithms. An incorrect threshold may result in loss of edges in the final edge map or may give rise to edges that are not in accord with the human visual perception. Ideally the threshold should be based on the local background and should adaptively change itself based on the background. In other words, the threshold selected should be dynamic and local [34] [35]. Probability models have been used in determining thresholds in the canny edge detection algorithm [36]. Methods have also been proposed for determining adaptive threshold based in the image histogram [37]. In this chapter we'll discuss adaptive thresholding that is based on the way a human observer views an object. Human visual system based thresholding results in an edge map that is more in accord with how human eye distinguishes objects from the background.

3.1 Review of the Human Visual Phenomenon

The human visual system (HVS) [38] is responsible for transferring data into information received by the viewer. The manner in which human eyes respond to visual stimulus is extremely complex and non-linear [39]. The received information is characterized by attributes like brightness, edge information, color shades etc. Brightness is actually a psychological sensation associated with the amount of light stimulus entering the eye. Brightness perceived is not a simple function of the intensity. Due to the great adaptive ability of the eye, human eye cannot measure the absolute brightness rather it measures the relative brightness. The relative brightness is an observer's feeling of difference in the grayness between the objects. This phenomenon can be observed even in our day to day experience. In a very dark room, even the slightest glimpse of light ray can cause a visible sensation in the eye whereas in a well illuminated room, the same amount of light may remain unnoticed. In other words, on a dark surface, human eye will be able

to detect small changes whereas on a bright surface, the change has to be of a substantial amount for the human eye to detect it. Also the sensitivity of the human eye to additive white noise depends on the background. Eyes are more sensitive to random noise in smooth areas of an image than in busy areas with a lot of texture or details. This is the underlying principle which guides the visual sensation of the human eye. The term contrast is used to emphasize the difference in luminance of objects [38] [40]. The perceived grayness of a surface depends on its local background and the perceived contrast remains constant if the ratios of contrasts between object and local background remain constant. The contrast C refers to the ratio of difference in luminance of an object B_o and its immediate surrounding B_s. Mathematically the contrast is given by,

$$C = \frac{|B_o - B_s|}{B_s} = \frac{\Delta B}{B_s} \tag{3.1}$$

The visual increment threshold (or just noticeable difference) is defined as the amount of light ΔB_T necessary to add to a visual field of intensity B such that it can be discriminated from a reference field of the same intensity B. At low light intensities, near absolute visual threshold, the luminance increment threshold is constant; then with increasing intensity, the threshold converges asymptotically to Weber behavior, i.e., $\Delta B_T \, \alpha \, B$. This type of behavior is exhibited in brightness incremental threshold for white broad-band spectra and monochromatic narrow-band spectra [16].

The characteristic response of the human eye is presented in Figure 3.1. The characteristic curve is represented in the $log\Delta B_T \; vs \; logB$ plane. The Weber behavior is generally expressed by the unit slope of the logarithmic curve [40]. The preceding region with slope 1/2 is known as the De Vries-Rose region. It has been shown that if the central visual processor behaves as an optimum

probabilistic detector, the incremental visual threshold follows the square root law, i.e. $\Delta B_T \, \alpha \, \sqrt{B}$. However, in the actual case this rule is followed in a small restricted region. The dashed curve shows the deviation from the Weber's law. This represents the saturation region. Though this behavior is not quite commonly exhibited by the retinal cone, yet we can expect this type of behavior in some restricted cases.

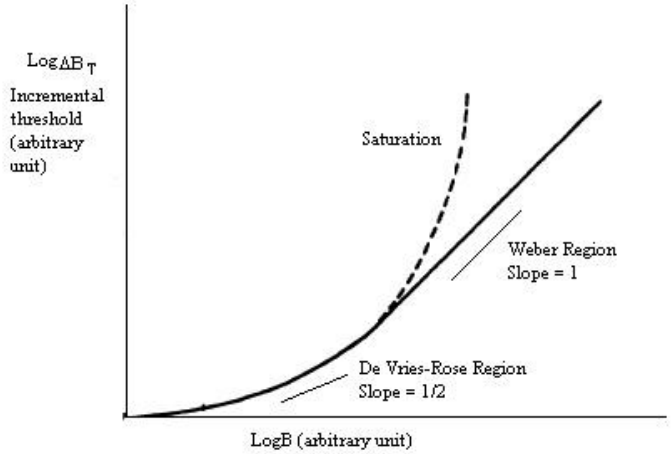


Figure 3.1: The Increment Threshold $\mathbf{\Delta B_T}$ as a function of Reference Intensity B

Hence it is evident that the variation of $log\Delta B_T$ against $logB$ in the De Vries-Rose region is slower than that in the Weber region. Therefore the discrimination ability of the human eye in the De Vries-Rose region is greater than that in the Weber region [41] [42] [43]. The possible reason for this deterioration in discrimination ability can be attributed to inherent visual nonlinearity.

To derive a mathematical model representing the inherent non-linearity of the human eye, we approximate the curve represented in Figure 3.1 by a piecewise linear curve as shown in Figure 3.2.

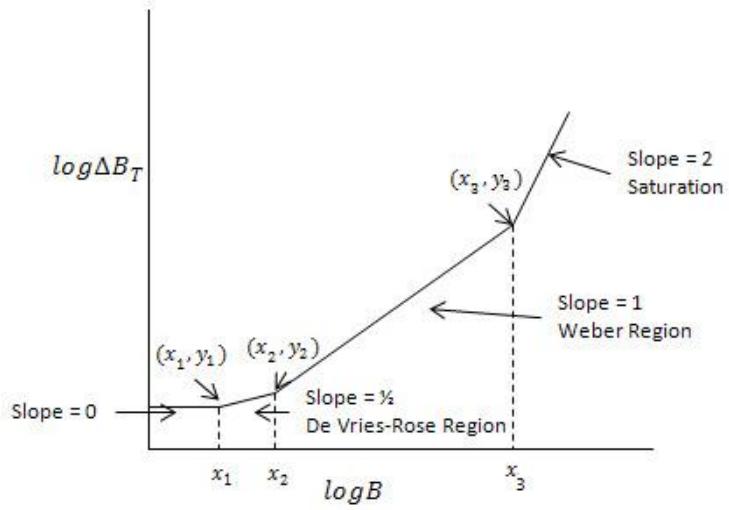


Figure 3.2: Linear approximation of the Increment Threshold $\mathbf{log\Delta B_T}$ as a function of Reference Intensity **logB**

The threshold values in the De Vries-Rose region, the Weber region and the saturation region are defined by the linear equations (3.2), (3.3) and (3.4) respectively [38] [44].

$$log\Delta B_T = \frac{1}{2} * \ logB + logK_2 \tag{3.2}$$

$$log\Delta B_T = logB + logK_1 \tag{3.3}$$

$$log\Delta B_T = 2 * \ logB + K_3 \tag{3.4}$$

Where K_1, K_2 and K_3 are constants.

Therefore, when the brightness value of an object is higher (or lower) than its surrounding or background or a reference intensity B by an amount $\geq \Delta B_T$, it corresponds to a point on or above the curve (Figure 3.2) and the object will appear either brighter (or darker).

The demarcations between the various regions are not so sharp and distinct. But for the sake of simplicity it is assumed that the De Vries-Rose region extends from x_1 to x_2, the Weber region extends from x_2 to x_3 and the saturation region extends beyond x_3. The value of B corresponding to $logB = x_i$ is assumed to be B_{x_i} for $i = 1,2,3$. Therefore,

$$x_i = logB_{x_i} \text{ for } i = 1,2,3 \tag{3.5}$$

Let x_0 and B_t be the maximum values of $logB$ and B respectively. The value of the parameter B_t which is the maximum difference in the image can be obtained by, $\mathrm{B_t} = \max\big(X(x,y)\big)\,\Theta\min(x(X,Y))$. The values of the $x_i's$ can then be expressed in terms of x_0 as,

$$x_i = \alpha_i x_0 \text{ for } i = 1,2,3 \tag{3.6}$$

$$\text{and } B_{x_i} = \alpha_i' B_t \text{ for } i = 1,2,3 \tag{3.7}$$

$$\text{where } 0 < \alpha_1 < \alpha_2 < \alpha_3 < 1,\ 0 < \alpha_1' < \alpha_2' < \alpha_3' < 1$$

Since the slope of the Weber region is 1, the ratio $\frac{\Delta B_T}{B}$ remains fairly constant over the entire Weber region with a value approximately equal to $\beta\%\ of\ \left(\frac{\Delta B_T}{B}\right)_{max}$. Hence from Equation 3.3 we can determine the value of K_1 as

$$K_1 = \frac{\Delta B_T}{B} = \frac{\beta}{100}\left(\frac{\Delta B_T}{B}\right)_{max} \tag{3.8}$$

At point (x_2, y_2) on the graph, both Equations 3.2 and 3.3 are satisfied. Therefore the value of K_2 can be deduced as,

$$x_2 + \ logK_1 = \ \frac{1}{2}x_2 + \ logK_2 \ \rightarrow \ \frac{1}{2}x_2 = \log\frac{K_2}{K_1} \rightarrow \frac{1}{2}logB_{x_2} = \log\frac{K_2}{K_1} \rightarrow K_2 = \ K_1\sqrt{B_{x_2}} \quad (3.9)$$

Similarly, at point (x_3, y_3) on the graph, both Equations 3.3 and 3.4 are satisfied. Therefore the value of K_2 can be deduced as,

$$K_3 = \frac{K_1}{B_{x_3}} \quad (3.10)$$

Hence the minimum values of the increment threshold for the different regions are given by,

$$\Delta B_T = \sqrt{B}\frac{\beta}{100}\left(\frac{\Delta B_T}{B}\right)_{max}\sqrt{B_{x_2}} \text{ in the De Vries-Rose region, } B_{x_2} \geq B \geq B_{x_1} \quad (3.11)$$

$$\Delta B_T = B\frac{\beta}{100}\left(\frac{\Delta B_T}{B}\right)_{max} \text{ in the Weber region, } B_{x_3} \geq B \geq B_{x_2} \quad (3.12)$$

$$\Delta B_T = B^2\frac{\beta}{100}\left(\frac{\Delta B_T}{B}\right)_{max}\frac{1}{B_{x_3}} \text{ in the saturation region, } B \geq B_{x_3} \quad (3.13)$$

Equations 3.11, 3.12 and 3.13 thus give the minimum visual increment thresholds that are required to distinguish an object from its immediate background. Hence for a pixel with intensity B_p, we should have

$$\text{either, } \quad \Delta B \geq \sqrt{B}\frac{\beta}{100}\left(\frac{\Delta B_T}{B}\right)_{max}\sqrt{B_{x_2}}, \quad \text{when} \quad B_{x_2} \geq B \geq B_{x_1} \quad (3.14a)$$

$$\text{or, } \quad \Delta B \geq B\frac{\beta}{100}\left(\frac{\Delta B_T}{B}\right)_{max} \quad \text{when} \quad B_{x_3} \geq B \geq B_{x_2} \quad (3.14b)$$

$$\text{or, } \quad \Delta B \geq B^2\frac{\beta}{100}\left(\frac{\Delta B_T}{B}\right)_{max}\frac{1}{B_{x_3}}, \quad \text{when} \quad B_{x_3} \leq B \quad (3.14c)$$

$$\text{where } \Delta B = \left|B_p - B\right|$$

for the point B_p to be treated as an edge pixel in the image. The gradient ΔB can be approximated by any of the edge detection algorithms. Thus HVS thresholding will decompose the image into four component images. These four component images can be subjected to any image processing algorithm and fused together to achieve better performance. The background intensity image B can be obtained by taking the local mean at each and every point in the image and is given by [11],

$$B(x,y)=\left[\ \mathrm{p} \otimes \left(\frac{\mathrm{p}}{2} \otimes \sum_{\mathrm{Q}} X(i,j) \oplus \frac{q}{2} \otimes \sum_{\mathrm{Q}'} X(k,l)\right) \oplus X(x,y)\right] \otimes \mathrm{p} \qquad (3.14d)$$

Here B(x,y) represents the background intensity at each pixel and X(x,y) is the input image. Q represents all the pixels that are directly left, right, up and down of the pixel and Q' is all of the pixels diagonally one pixel away. Also p and q are constants. Let us assume that the input image is $X(x,y)$ and the gradient image is $X'(x,y)$. Then the four HVS based sub-images are given by,

$$Im1 = X(x,y)\ for \quad B_{x_2} \geq B(x,y) \geq B_{x_1}\ \&\ \frac{X'(x,y)}{\sqrt{B(x,y)}} \geq K_2 \qquad (3.15a)$$

$$Im2 = X(x,y)\ for \quad B_{x_3} \geq B(x,y) \geq B_{x_2}\ \&\ \frac{X'(x,y)}{B(x,y)} \geq K_1 \qquad (3.15b)$$

$$Im3 = X(x,y)\ for \quad B_{x_3} \leq B(x,y)\ \&\ \frac{X'(x,y)}{B(x,y)^2} \geq K_3 \qquad (3.15c)$$

$$Im4 = X(x,y)\ for\ all\ remaining\ pixels \qquad (3.15d)$$

3.1.1 Image Decomposition based on HVS

In this section we present the results of image decomposition by applying HVS based non-linearity. Figure 3.3 shows a grayscale image having varying background intensities. According to Figure 3.2, the background image (which can be derived by taking the local background in

each and every point in the image) can be divided into four sub-images according to their intensities based on the HVS. Figure 3.4 shows the different background images in the different regions. In Figure 3.5 we show the ideas expressed in Equation 3.14 (a-c). The gradient of an image pixel (i.e. the difference between the pixel intensity with its immediate background) is now a function of the background intensity also. Hence four different sub-images are possible in the four regions according to the human visual response.

Figure 3.3: Original Grayscale Image

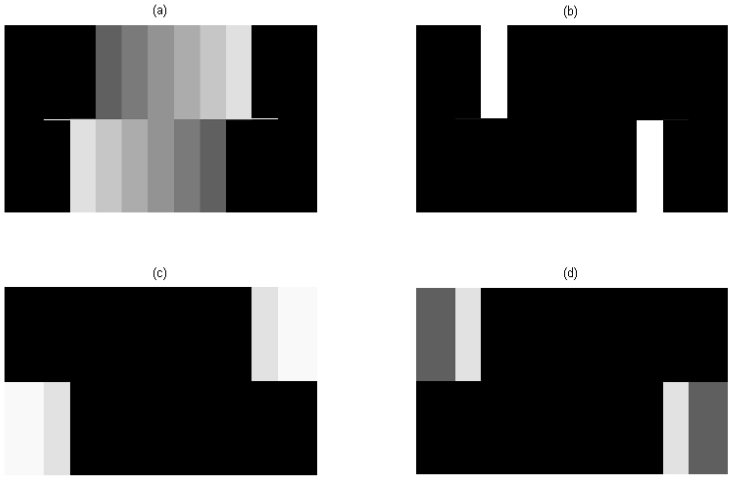


Figure 3.4: HVS based background images in (a) Weber region (b) De Vries-Rose Region (c) Saturation region and (d) Dark region

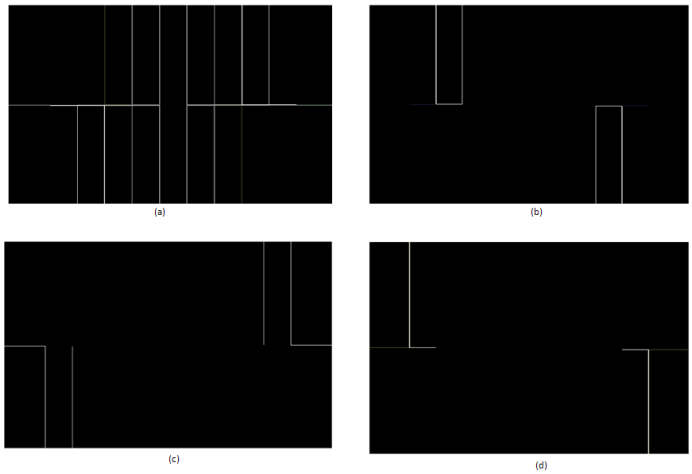


Figure 3.5: HVS based Gradient in (a) Weber region (b) De Vries-Rose Region (c) Saturation region and (d) Dark region

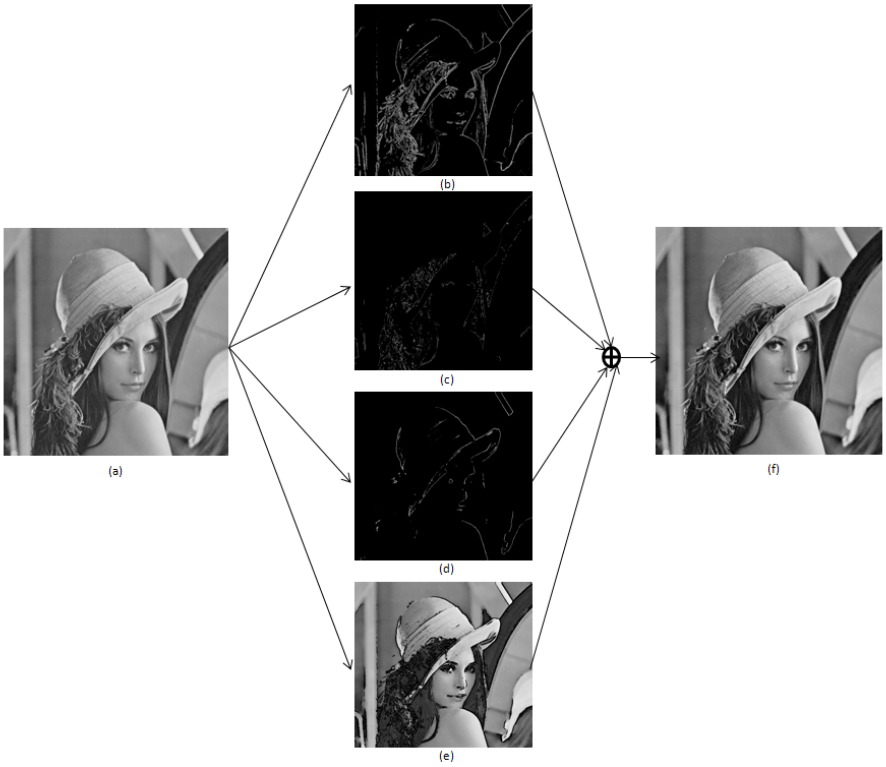


Figure 3.6: (a) Original Image, HVS based Images in (b) Weber Region (c) De Vries-Rose Region (d) and Saturation region, (e) Remaining image pixels (Other region) and (f) Result of arithmetic addition of (b), (c), (d) and (e)

We show the decomposition of an image into four sub-images according to the human visual non-linearity according to Equation 3.15 (a-d). Figure 3.6 shows the HVS based decomposition of an image and how the result after the four sub-images are fused together using arithmetic addition.

3.1.2 HVS based edge detection

In this section we'll show examples of edge detection based on HVS. The goal of HVS is to decompose the image into different regions based on background intensity and then apply a threshold that is based on the background intensity. Four different edge maps are obtained based on the background intensity. The edge maps are then fused to get the final HVS based edge map.

Figure 3.6 and 3.7 shows the edge extraction examples using HVS based edge detection.

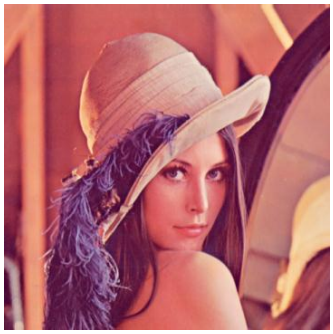


Figure 3.6.1: Original Image

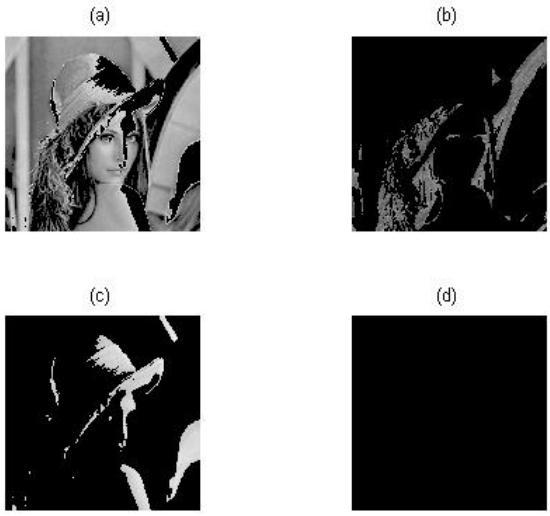


Figure 3.6.2: HVS based background images in (a) Weber region (b) De Vries-Rose Region (c) Saturation region and (d) Dark region

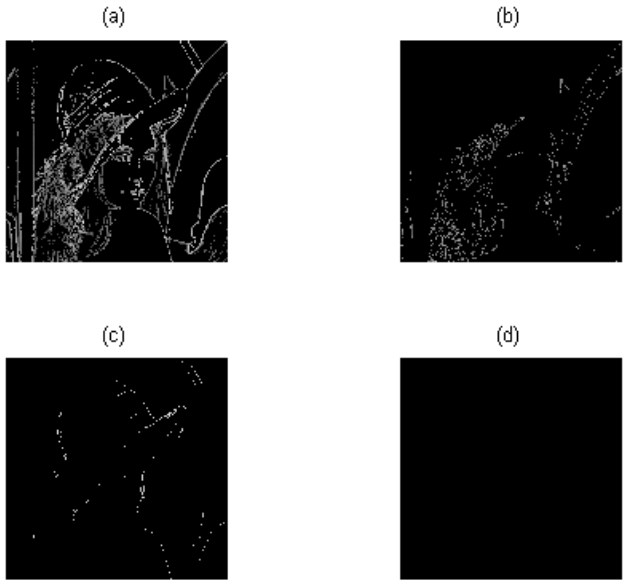


Figure 3.6.3: HVS based Gradient in (a) Weber region (b) De Vries-Rose Region (c) Saturation region and (d) Dark region

Figure 3.6.4: Edge map obtained by HVS

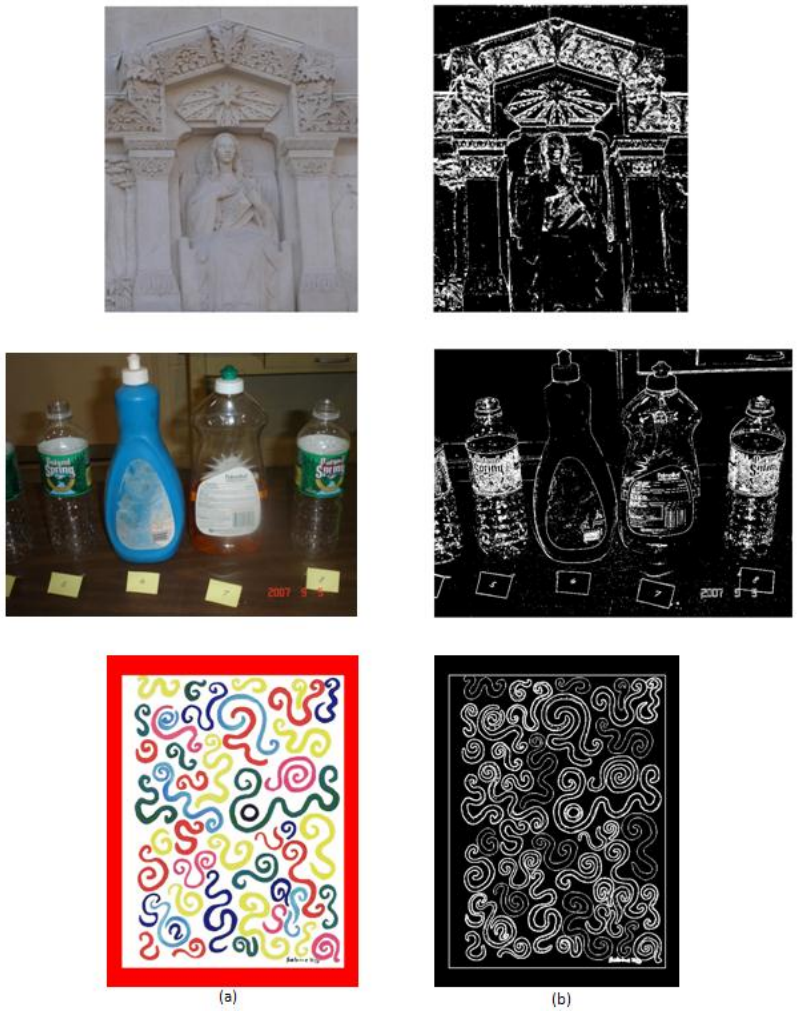


Figure 3.7: (a) Original Images (b) Results of edge detection by HVS

3.2 Parameter Selection for HVS

Two sets of parameters dictate the edge detection and image decomposition algorithms using HVS. The alpha parameter set determine the boundaries of the different regions in HVS decomposition. The parameter beta determines the vertical height of the visual increment threshold curve given in Figure 3.2. In this section of the thesis we will discuss the experimental results obtained by alpha and beta variation and their effect on the edge detection and image decomposition algorithms. We'll present the effects of parameter variation on both synthetic and natural images.

3.2.1 Alpha variation

The alpha parameters determine the boundary of the different regions for HVS based image decomposition as it is evident from Equations 3.6 and 3.7. The Weber region is given the maximum weightage and hence this region is the largest compared to the other regions in the HVS based image decomposition. The typical values chosen for alpha parameters are: $\alpha 1 = 0.1$, $\alpha 2 = 0.3$ and $\alpha 1 = 0.9$ for edge detection and $\alpha\ 1 = 0$, $\alpha 2 = 0.1$ and $\alpha 1 = 0.9$ for image decomposition. The selection of alpha values depends on the application itself and the values need to be adjusted depending on how much weightage needs to be placed on the individual regions. For example if an image has areas of shadows, the span of the De-Vries Rose region is increased. However, the Weber region gets the maximum weightage almost always.

We present in this section the results of HVS based image decomposition for 2 sets of values of the alpha parameter for a constant value of the parameter beta.

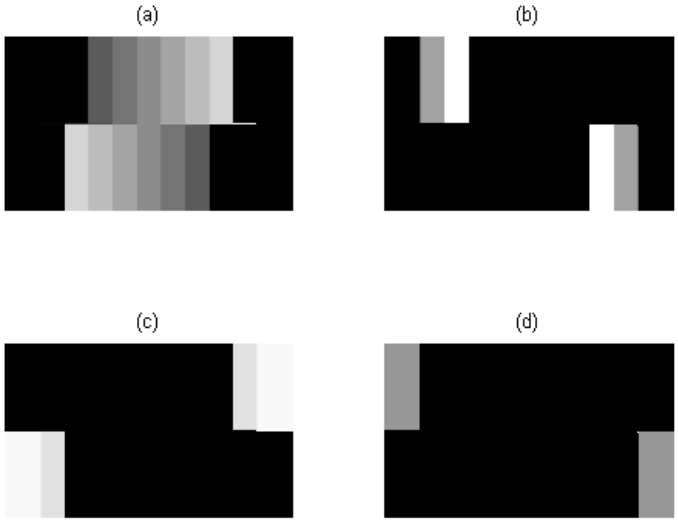


Figure 3.8: HVS based background images in (a) Weber region (b) De Vries-Rose Region (c) Saturation region and (d) Dark region for $\boldsymbol{\alpha 1 = 0.1}$, $\boldsymbol{\alpha 2 = 0.3}$ and $\boldsymbol{\alpha 1 = 0.9}$

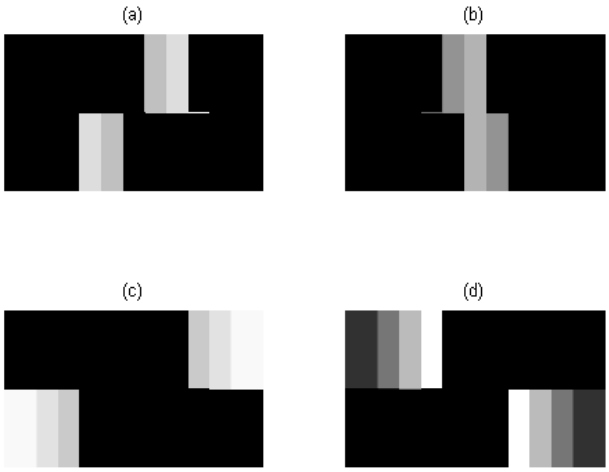


Figure 3.9: HVS based background images in (a) Weber region (b) De Vries-Rose Region (c) Saturation region and (d) Dark region for $\boldsymbol{\alpha 1 = 0.4}$, $\boldsymbol{\alpha 2 = 0.6}$ and $\boldsymbol{\alpha 1 = 0.8}$

In Figure 3.8, the majority of the information lies in the Weber region as the Weber region is the largest in this case. In Figure 3.9, the amount of information in the Weber region is much reduced due to the reduced weightage in this region. Figure 3.10 and 3.11 shows the edge map obtained corresponding to the alpha values in Figure 3.8 and 3.9.

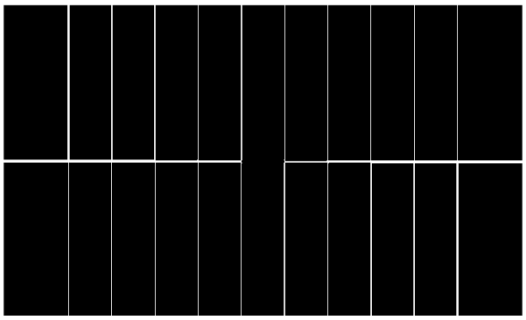


Figure 3.10: Edge map corresponding to $\boldsymbol{\alpha 1 = 0.1}$, $\boldsymbol{\alpha 2 = 0.3}$ and $\boldsymbol{\alpha 1 = 0.9}$

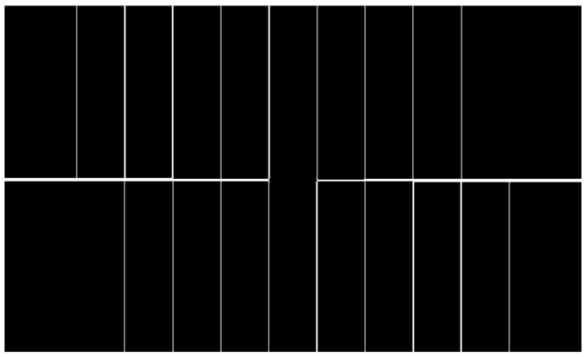


Figure 3.11: Edge map corresponding to $\boldsymbol{\alpha 1 = 0.4}$, $\boldsymbol{\alpha 2 = 0.6}$ and $\boldsymbol{\alpha 1 = 0.8}$

Figure 3.11 has lost some edge information in the saturation region, whereas the edges corresponding to the other regions remain unaffected. HVS based thresholding is based on the underlying principle that the visual increment threshold increases with the increase of the background intensity and the rate of change depends on the region in which the background

intensity belongs. The rate of change is the highest in the saturation region followed by the Weber, De-Vries Rose and the dark regions. Hence alpha variation has less effect on edge detection by HVS. Experimentally it has been found that giving the maximum weightage to the Weber region yields best results.

3.2.2 Beta variation

Beta is the thresholding parameter for HVS based image processing algorithms. We recollect the equations (3.2), (3.3) and (3.4) here and see that each of the equations is in the form of the linear equation $y = mx + c$, where m is the slope and c is the intercept. The constants K_1, K_2 and K_3 determine the magnitude of the positive intercept. If the intercept is positive, then increasing its value shifts the straight line up. The constants K_1, K_2 and K_3 are derived from the parameter beta and hence if beta is increased, the visual increment threshold curve moves in the direction of positive y-axis thereby increasing the threshold for HVS decomposition.

Effect of beta variation for edge detection

The choice of the parameter beta plays a very important role in edge detection by HVS. A very low value of beta may result in false edges appearing in the edge map. Increasing the value of beta to a great extent may result in loss of edges. The value of beta also depends on the choice of the image. The effect of beta variation is illustrated below. We fix the alpha parameter values at $\alpha 1 = 0.1$, $\alpha 2 = 0.3$ and $\alpha 1 = 0.9$ giving the maximum weightage to the Weber region.

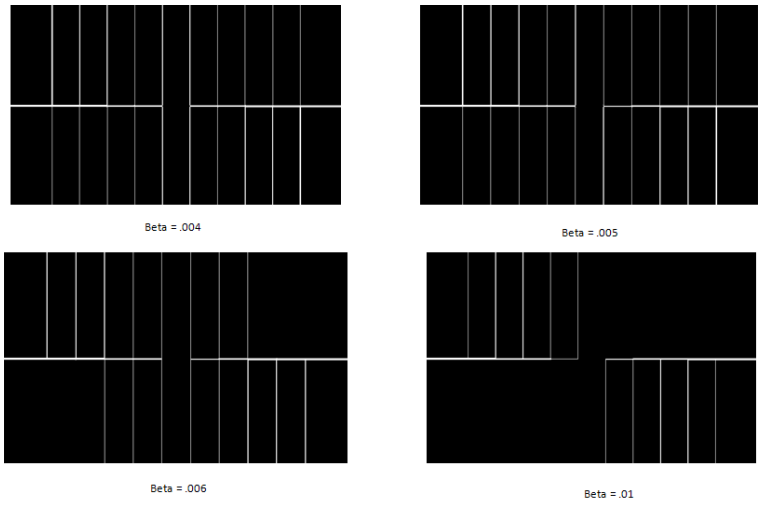


Figure 3.12: Effect of beta variation on edge map

From figure 3.12 we can see that as the value of beta increases, there is loss of edge in the high intensity regions because as the background intensity increases, the rate of change of threshold also increases. Thus we see that synthetic images need a very low value of beta.

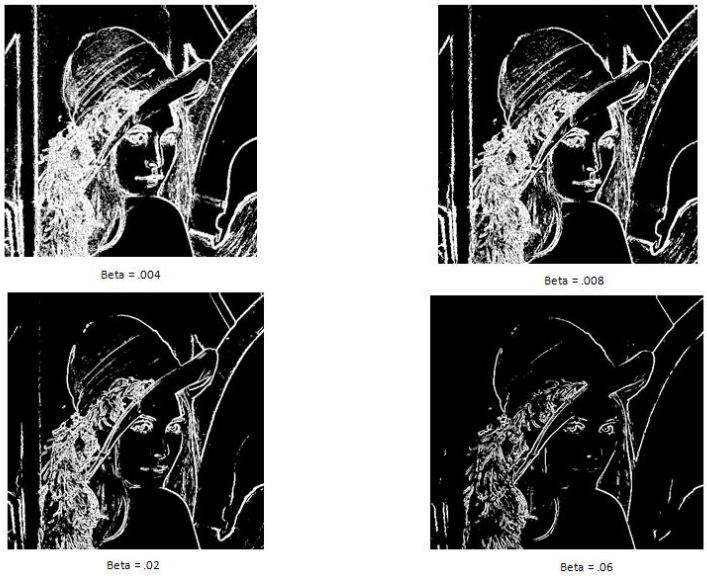


Figure 3.13: Effect of beta variation on edge map

The effect of change of beta is also noticed in the image of "Lena". At low values of beta, false edges appear in the images. A value of .02 for the parameter gives the best edge map in this case. As the value of beta is increased further, edges are lost. Thus for natural images the value of beta should be kept at mid-range (0.01 – 0.06).

Effect of beta variation for image decomposition

In this section we analyze the effect of beta variation on image decomposition. From equation 3.15 (a) through (d), it is evident that the images in the four HVS decomposed regions are complementary and the original image can be retrieved by adding the four regions together. The Weber, De-Vries Rose and Saturation regions represent areas in the image when the gradient magnitude is perceivable by the human eye. These regions collectively represent the high frequency regions in an image. The fourth region or the "Other" region represents the low frequency region where the gradient magnitude remains constant according to the human eye. The thresholding parameter beta controls the amount of information to be placed in the Other region as compared to the information to be placed on the high frequency regions. A low value of beta place less information in the Other region whereas a high value places more information in the Other region compared to the Weber, De-Vries Rose and Saturation regions. This is useful when we are using the Other region alone in certain applications. We'll fix the alpha parameter values at $\alpha 1 = 0.1$, $\alpha 2 = 0.3$ and $\alpha 1 = 0.9$ and show the effect of changing the values of the parameter beta. We'll select three value of beta; a value equal to 0.002 in the low range; 0.02 in the mid-range; and 0.2 in the high-range.

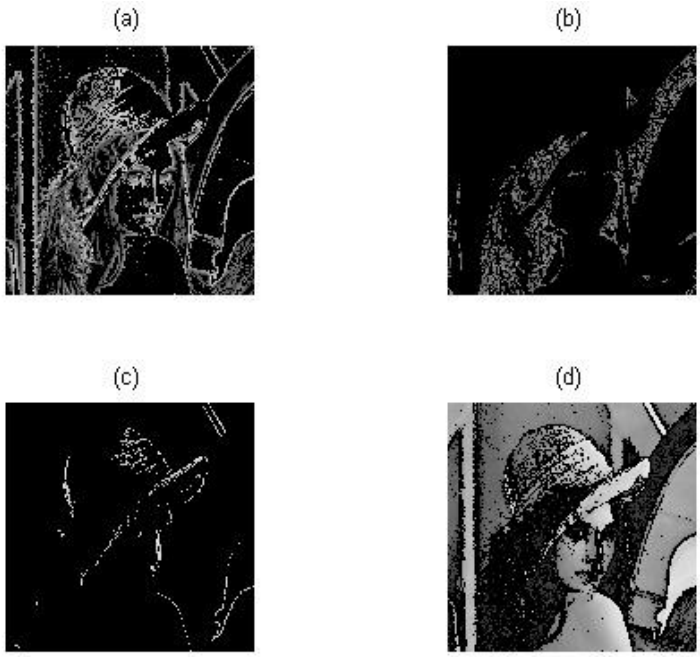


Figure 3.14: HVS decomposed images in (a) Weber Region (b) De-Vries Rose Region (c) Saturation region and (d) Other region for beta = 0.002.

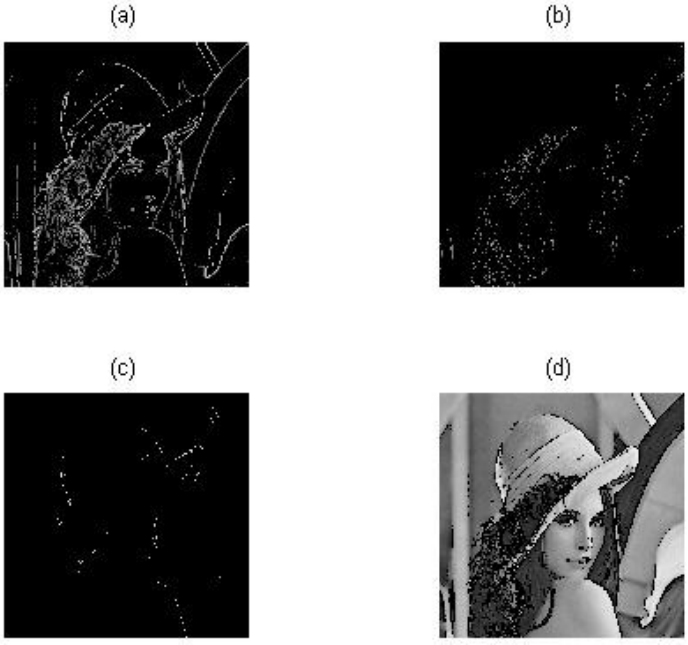


Figure 3.15: HVS decomposed images in (a) Weber Region (b) De-Vries Rose Region (c) Saturation region and (d) Other region for beta = 0.02.

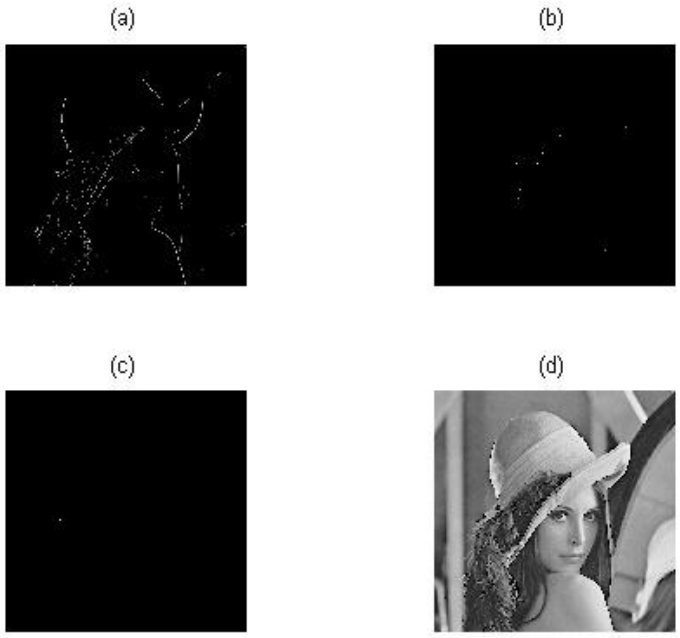


Figure 3.15: HVS decomposed images in (a) Weber Region (b) De-Vries Rose Region (c) Saturation region and (d) Other region for beta = 0.2.

Range of beta	Typical Values	Applications
Low	0.001 – 0.005	Edge detection in synthetic images
Medium	0.02 – 0.06	Edge detection in natural images, edge based feature extraction from natural images and other applications involving the gradient image.
High	0.2-0.4	Amount of gradient information to be retained is really low. Not used much

Table 3.1: Summary of beta values

Table 3.1 summarizes the typical values of beta for different applications.

CHAPTER 4

REVIEW OF FEATURE EXTRACTION FROM IMAGES

Computer vision techniques are widely used nowadays to address real world problems that involve processing of voluminous image data requiring massive computing efforts and high level of processing accuracy. Such requirements are addressed using sophisticated form of image processing applications. Computer vision systems are applied to solve complex object recognition and classification problems in medical and security applications. For example, object recognition techniques are used in medical applications for breast and prostate cancer detection. Such applications use MRI scan images of the patients and assist the medical team to identify the presence of tumors. A medical tumor classification system provides useful information about the tumor's nature, malignity etc. and help the medical team to plan effective diagnosis procedures. In security and automatic surveillance systems computer vision techniques are widely used to identify threat objects like guns or other weapons from scanned images. Face detection and recognition systems are heavily based on the computer vision techniques. Detecting eyes from facial images is another application where these techniques are applied. It has been found that a majority of car accidents are caused by driver fatigue. Hence automated systems have been developed that can track the eyes of the driver in subsequent frames of images and warn the driver in case the eyes remain closed for a subsequent amount of time. Pedestrian detection for security surveillance is also based on computer vision techniques. All these methods are fundamentally based on training the system based on the application and then characterize a new object based on the training set. Ideally the raw image can be used to train the system. However considering the typical size of a two dimensional or three dimensional images, this would result in a huge volume of data that would render the system impractical for real time applications.

Hence suitable object characterization techniques are used widely to reduce the parameters needed to characterize an object.

In computer vision applications, feature extraction is used widely as a method of object characterization. A human observer can easily identify and classify objects in an image from his experience and past knowledge gathered over the years. For example a human observer can easily identify a face in an image even under the influence of shadow or partial occlusion. However the task becomes extremely challenging for an automated computer vision application. Hence objects are generally represented as a vector of numerical features called the feature vector. Each of the values in the vector is called a feature and describes a specific aspect of the object. For instance, a feature may describe an object's physical properties like area, perimeter, volume, surface areas and so on. The primary goal behind this object representation technique is to reduce the dimensionality of the problem by generating a compressed object representation that will result in less processing efforts and memory requirements. The efficiency of the classification process is controlled by the quality of the feature vectors. Several feature vector methods are available to address recognition and classification problem. Feature vectors can be based on the binary edge map, gradient image or simply the intensity values in the image. It has been found that the selection of feature vectors used also depend widely on the application in which they are used.

Object detection and recognition systems often fuse more than one feature vector algorithms for efficient detection and recognition. In this chapter will discuss some of the state of the art feature vector extraction algorithms.

4.1 The Cell Edge Distribution (CED)

The Cell Edge Distribution (CED) [45] [46] [47] [48] [49] is method of edge based feature extraction from an image. The image is first subjected to edge detection for extracting edge maps in the horizontal, vertical, diagonal and anti-diagonal directions. Here we have used HVS for obtaining the directional edge maps. In the next step each edge map is divided in 16 equal rectangular or square cells of equal size depending on the size of the image. Feature maps are generated by counting the number of edge pixels per cell. Thus each directional edge map generates a vector of sixteen feature values. The feature maps generated in the horizontal, vertical, diagonal and anti-diagonal directions are concatenated together to formulate a 64 bit feature vector.

The CED method of feature extraction consists of the following steps:

1. Get the input image in grayscale
2. Apply directional kernels for extracting gradients in the horizontal, vertical, diagonal and anti-diagonal directions
3. Threshold the 4 gradient images for extracting the 4 directional edge maps using HVS
4. Divide each of the edge maps into 16 cells

Cell1	Cell2	Cell3	Cell4
Cell5	Cell6	Cell7	Cell8
Cell9	Cell10	Cell11	Cell12
Cell13	Cell14	Cell15	Cell16

5. Count the edge flags in each of the cells. Thus each of the edge maps will give rise to a 16 element feature vector.
6. Fuse the four 16-element feature vectors to get the final feature vector consisting of 64 elements.

The size of the directional gradients can be chosen based on application. Typically 3X3 or 5X5 kernels are used. Threshold can be chosen to global or local. Human Visual System based thresholding has been applied for our experiments. The CED method for feature extraction is easy to implement and works well for security applications for gun or liquid bottle detection. This has also been for face detection by fusion of CED with other feature vectors like the Principal Projected Edge Distribution (PPED) described in the next section of this chapter.

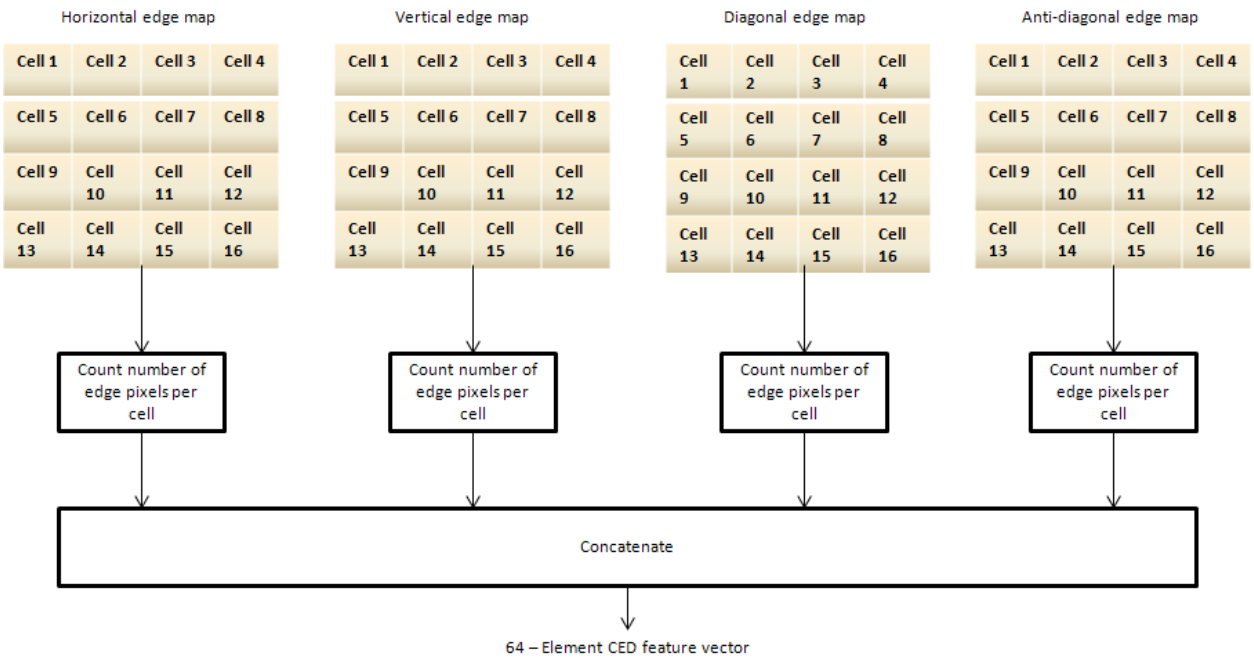


Figure 4.1: Schematic Diagram of the CED feature extraction algorithm

Figure 4.2.1: Original example image

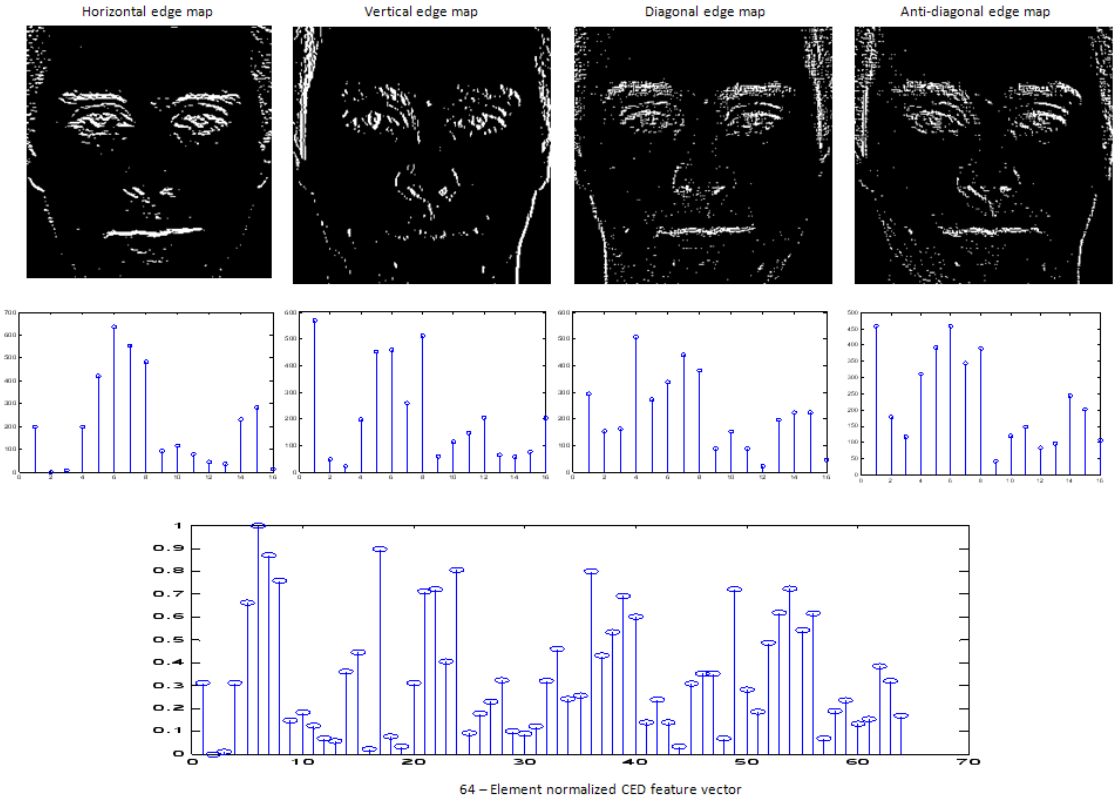


Figure 4.2.2: CED feature vector extracted from image in Figure 4.2.1. Edge maps obtained by HVS edge detection

4.2 The Principal Projected Edge Distribution (PPED)

The principal projected edge distribution (PPED) [45] [46] [49] [50] is another approach towards edge based feature extraction in images. PPED feature vectors are often fused with the CED feature vectors for object characterization. They combined feature vector has been used for face extraction from images. Like the CED feature vector, PPED also starts by extracting the directional edge maps based on HVS from the images. Once again human visual system based thresholding is used for obtaining the edge maps. However in this case the feature maps are divided in the same direction as that of the edge. The horizontal edge map, for example, is cut along the rows; the vertical edge map is cut along the columns and so on. The edge flags in the cells are then collected from the projected divisions and concatenated as before to get the final 64 element feature vector.

The Mojette transform [51] is a discrete version of the Radon transform which gives projections of an image along different orientations. A projection image defines the image nature at a specific angle. A projection oriented at an angle θ_i for an image f is given by,

$$proj_{pi,qi}(b) = \sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} f(k,l)\Delta(b + kq_i - lp_i) \tag{4.1}$$

where θ_i is related to p_i and q_i by $\theta_i = \tan^{-1}\left(\frac{q_i}{p_i}\right)$ and $\Delta(b)$ is the Kronecker delta function which is 1 for $b = 1$ and 0 otherwise. A projection sums the pixel value of pixels which cross the line $b = lp_i - kq_i$ for every combination of k and l. The Mojette transform is the set of projections for I predetermined projection angles. PPED feature extraction is essentially the Mojette transform with $(p_i, q_i) = (1,0)$ for a horizontal edge map, $(p_i, q_i) = (0,1)$ for a vertical edge map, $(p_i, q_i) = (1,1)$ for a +45 degree edge map, and $(p_i, q_i) = (1,-1)$ for a -45 degree

edge map, except that PPED feature extraction counts the number of edge pixels in some specific number (depending on the size of the image) of rows along the orientation direction of an edge map rather than every row along the orientation direction of an edge map.

The steps involved in the PPED feature vector extraction algorithm can be summarized as follows:

1. Get the input image in grayscale
2. Apply directional kernels for extracting gradients in the horizontal, vertical, diagonal and anti-diagonal directions
3. Threshold the 4 gradient images using HVS for extracting the 4 directional edge maps
4. Divide the image depending on the edge orientation
5. Count the number of edge flags in each division to generate the feature vectors from the individual edge maps.
6. Concatenate the feature vectors generated in step 5 to get the combined PPED feature vector.

The PPED feature vector extraction algorithm however restricts the input image to be a square image.

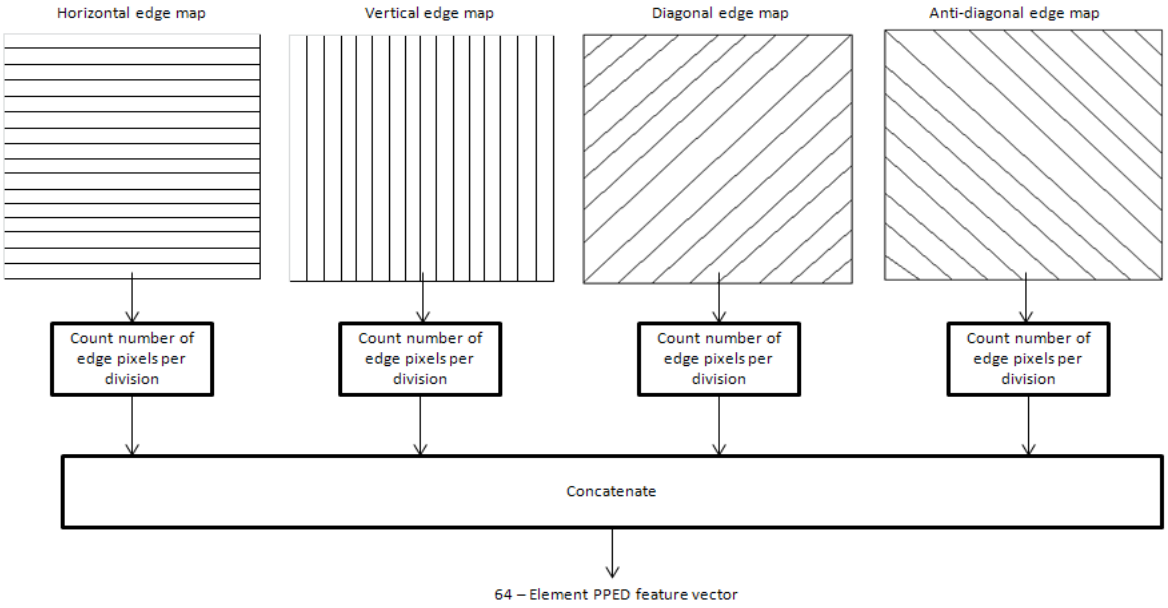


Figure 4.3: Schematic Diagram of the PPED feature extraction algorithm

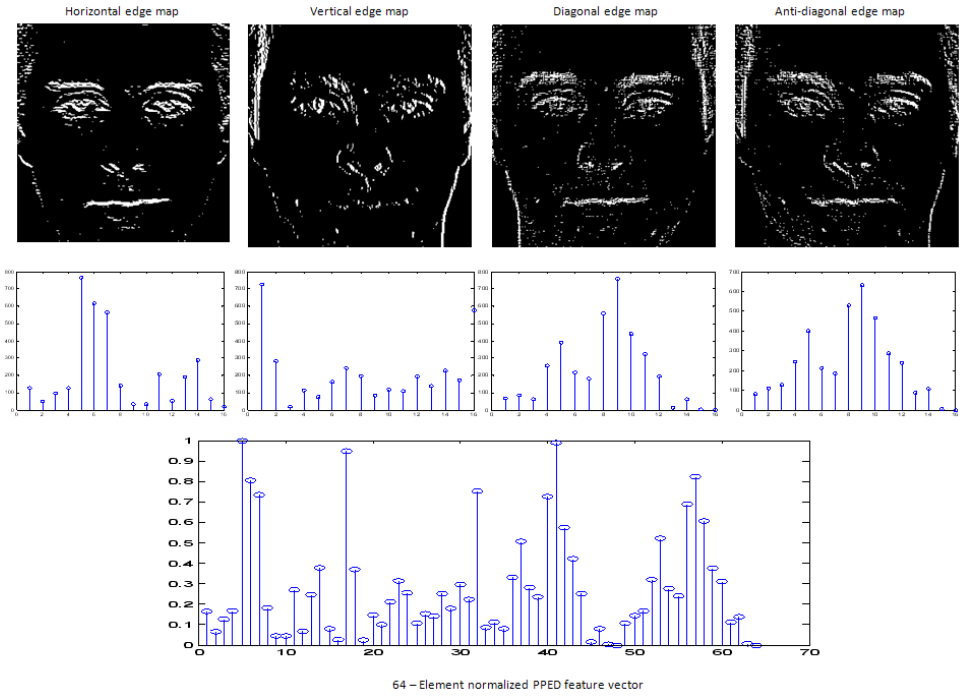


Figure 4.4: PPED feature vector extracted from image in Figure 4.2.1. Edge maps obtained by HVS

4.3 Feature Fusion using CED and PPED

As it has been mentioned in the previous section, the CED and PPED feature vectors are often fused together for object characterization. CED feature extraction is generally more robust than PPED feature extraction. However, PPED feature vectors contain very useful information, especially when using smaller windows. With that said, the minimum window size on which CED can be performed is 4x4 and the minimum window size on which PPED can be performed is 16x16 simply due to the nature of the divisions. A simple strategy for fusion that has been used for face detection combines the 2 feature vectors together to form 128 element feature vector. An alternative way of feature fusion has been proposed in [52]. It has been used for feature extraction is security applications like gun detection. The only difference between the CED and PPED methods is the shape and arrangement of the divisions in which edge pixels are counted in. A desirable way of fusing these two feature vectors should be such that the desirable properties of both of the feature vectors are retained. The shape and arrangement of the divisions in which edge pixels are counted in has been illustrated in Figure 4.5. This type of feature extraction method incorporates the qualities of CED and PPED simultaneously. CED feature extraction is performed on the majority of the edge map since it is generally a more robust method, and a modified PPED feature extraction is performed on a smaller window located in the center of the edge map where it will work more effectively.

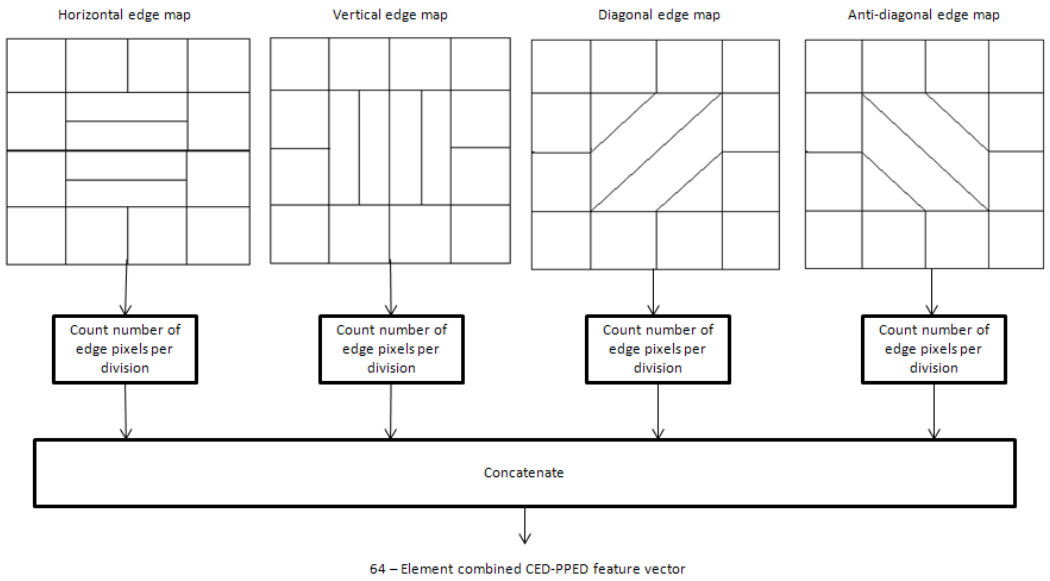


Figure 4.5: Schematic Diagram of the CED-PPED feature extraction algorithm

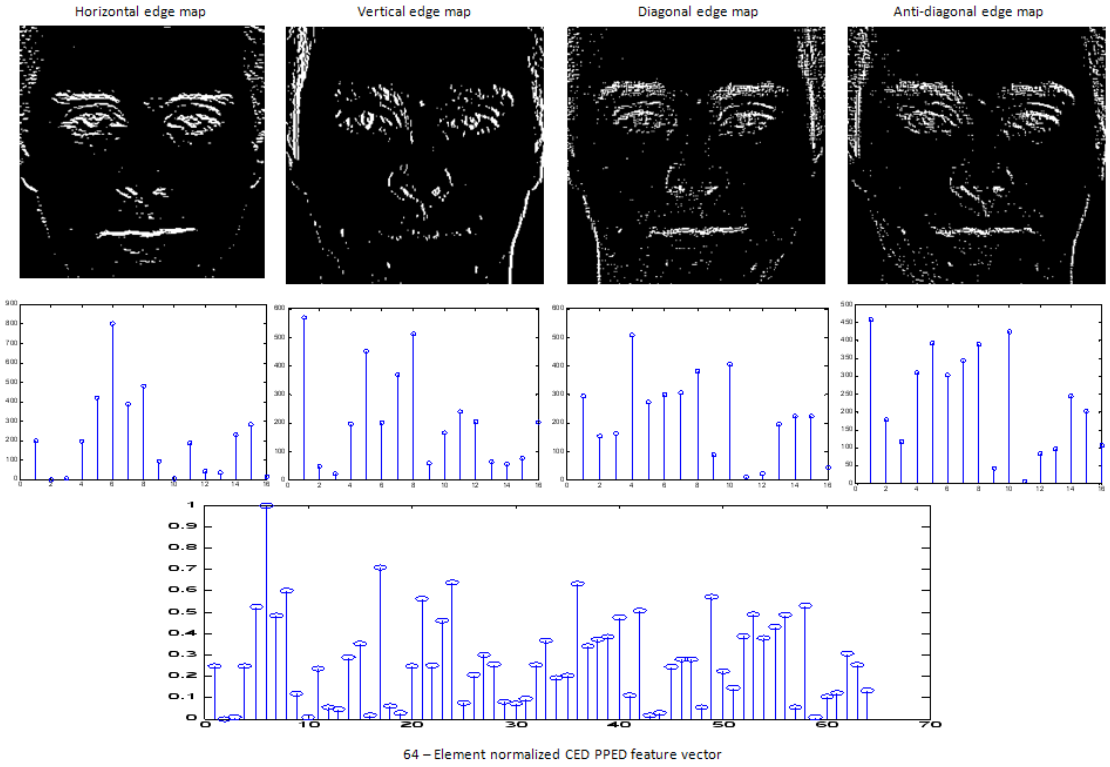


Figure 4.6: Combined CED-PPED feature vector extracted from image in Figure 4.2.1. Edge maps obtained by HVS

4.4 Histogram of Oriented Gradients (HOG)

The histogram of oriented gradients (HOG) [53] [54] [55] [56] [57] is a method of feature extraction that is based on the gradient of the image. HOG is based on the Edge Orientation Histograms (EOH) [58]. The basic idea of EOH is to get the edge orientation angles from the gradient image and then compute the histogram of the orientation angles. The orientation analysis gives robustness to lighting changes whereas the histogram gives translational invariance. HOGs are computed on a dense grid of uniformly spaced cells and they use overlapping local contrast normalizations for improved performance.

HOG algorithm for feature extraction is implemented by dividing the image window into small spatial regions ("*cells"*), for each cell accumulating a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell. The combined histogram entries form the representation. For better invariance to illumination, shadowing, etc., it is also useful to contrast-normalize the local responses before using them. This can be done by accumulating a measure of local histogram energy over somewhat larger spatial regions ("*blocks"*) and using the results to normalize all of the cells in the block. The normalized descriptor blocks are known as the HOG descriptors. The HOG representation has several advantages. It captures edge or gradient structure, and it does so in a local representation with an easily controllable degree of invariance to local geometric and photometric transformations: translations or rotations make little difference if they are much smaller that the local spatial or orientation bin size. The HOG algorithm is one of the state of the art methods for human detection in various surroundings.

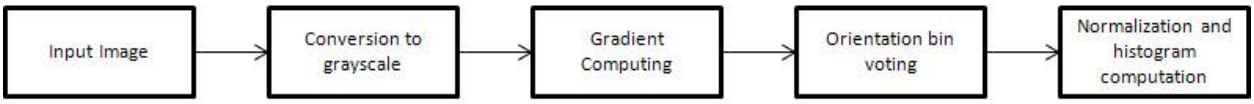


Figure 4.7: Schematic diagram of the HOG feature extraction algorithm

The steps involved in the HOG feature extraction algorithm can be summarized as follows:

1. Get the input image and convert it to grayscale

2. Compute the gradient image: Gradients can be computed by using Gaussian smoothing followed by spatial filtering of the image with any of the discrete derivative operators.

3. Spatial/Orientation Binning: In this step each pixel calculates a weighted vote for an edge orientation histogram channel based on the orientation of the gradient element centered on it, and the votes are accumulated into orientation bins over local spatial regions (cells). Cells can be either rectangular or radial. The orientation bins are evenly spaced over $0° - 180°$ (unsigned gradient) or $0° - 360°$ (signed gradient).The vote is a function of the gradient magnitude at the pixel, either the magnitude itself, its square, its square root, or a clipped form of the magnitude representing soft presence/absence of an edge at the pixel.

4. Normalization over HOG descriptor blocks: Gradient strengths vary over a wide range owing to local variations in illumination and foreground-background contrast. Hence effective local contrast normalization is essential for good performance. The HOG

descriptor blocks are obtained by grouping cells into larger spatial blocks and contrast normalizing each block separately. The final descriptor is the vector of all components of the normalized cell responses from all of the blocks in the detection window. Typically blocks are also overlapped. Two classes of block geometries are usually used, square or rectangular ones partitioned into grids of square or rectangular spatial cells (R-HOGs), and circular blocks (C-HOGs) partitioned into cells in log-polar fashion. The different block normalization schemes that are generally used for HOG descriptors include $L_1 - norm, L_2 - norm, L_2 - Hys$ and $L_1 - sqrt$. Let v be the un-normalized vector. Then its k^{th} norm is given by $\|v\|_k$ where $= 1,2,3, \ldots$. The $L_1 - norm$ normalization is $v/(\|v\|_1 + \epsilon)$ where ϵ is a small constant. $L_2 - norm$ normalization is given by $v/\sqrt{(\|v\|_2^2 + \epsilon^2)}$. $L_1 - sqrt$ norm is given by $v/\sqrt{(\|v\|_1 + \epsilon)}$. $L_2 - Hys$ is $L_2 - norm$ followed by clipping (limiting maximum values of) v to 0.2).

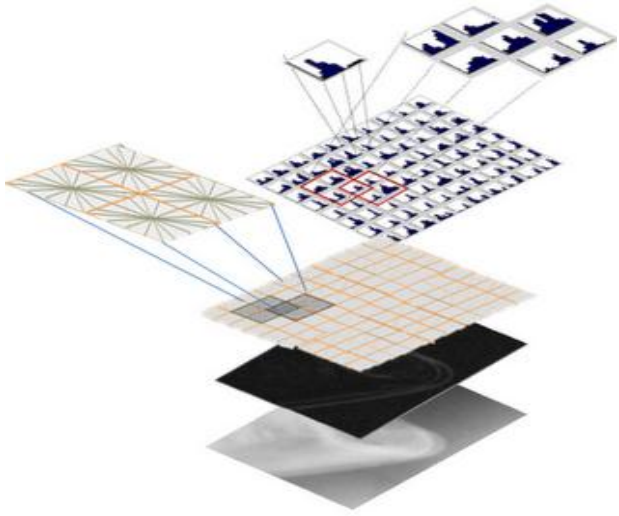


Figure 4.8: Representation of the HOG feature descriptor algorithm

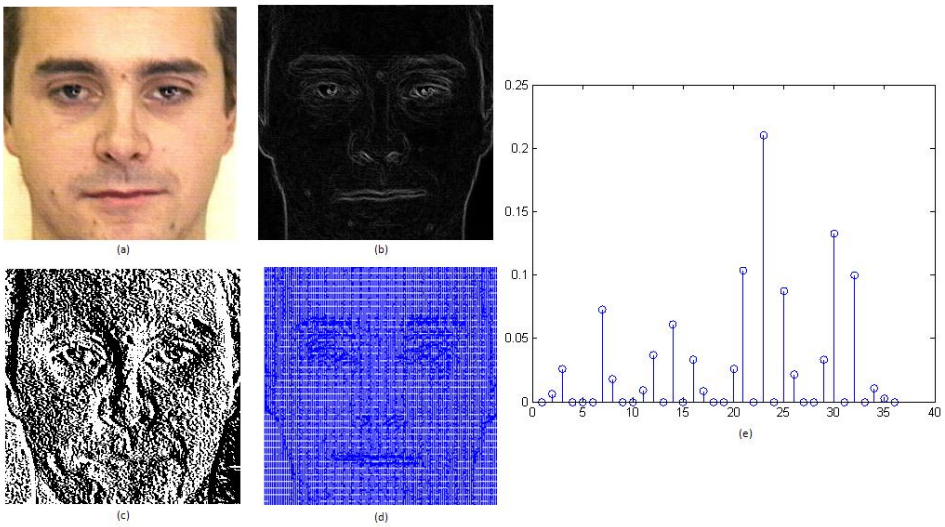


Figure 4.9: (a) Original Image (b) Gradient Magnitude (c) Gradient Angle (d) Gradient vector at each point retains the features of the facial image (e) 36-element HOG Feature vector extracted from one R-HOG block.

Complete feature vector is obtained by concatenating the feature vectors for each block. The number of blocks depends on the size of the image, the number of cells in the block and the amount of overlapping allowed between the adjacent blocks.

4.5 Local Binary Patterns (LBP)

Local Binary Patterns (LBP) is an image feature [59] based on the intensity values in the image. LBPs were originally used as a texture descriptor in an image. However they can be extended for describing facial features and have been applied successfully for face recognition [60] [61] [55] [56]. It is highly discriminative and is invariant to monotonic gray level changes. LBPs have also been used successfully for face recognition.

The LBP operator assigns a label to every pixel of an image by thresholding a 3x3-neighborhood of each pixel with the center pixel value and considering the result as a binary number. Then the histogram of the labels can be used as a texture descriptor.

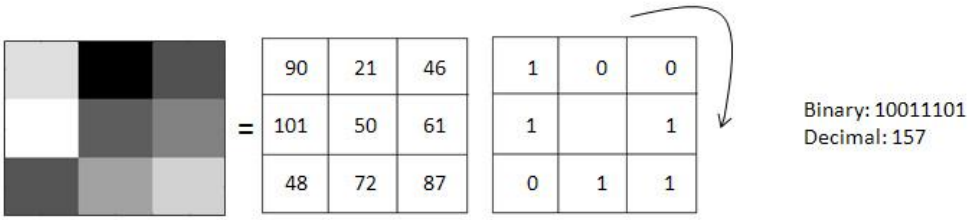


Figure 4.10: Basic LBP operator

The neighborhoods of the LBP operator can be various sizes. The local neighborhood can be defined as a set of sampling points evenly spaced on a circle centered at the pixel to be labeled. Thus any radius and any number of sampling points can be used for feature description. Bilinear interpolation is used when a sampling point does not fall in the center of a pixel. The neighborhoods are represented using the notation (P, R) where P denotes the set of points on a circle of radius R.

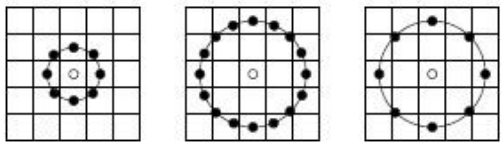


Figure 4.11: Circular (8, 1), (16, 2) and (8, 2) neighborhood

An extension to the original LBP operator is the *uniform local binary patterns*. A local binary pattern is called uniform if the binary pattern contains at most two bitwise transitions from 0 to 1 or vice versa when the bit pattern is considered circular. For example, the patterns 00000000 (0 transitions), 01110000 (2 transitions) and 11001111 (2 transitions) are uniform whereas the patterns 11001001 (4 transitions) and 01010011 (6 transitions) are not. In the computation of the

LBP histogram, uniform patterns are used so that the histogram has a separate bin for every uniform pattern and all non-uniform patterns are assigned to a single bin. This is due to the fact that uniform patterns account for about 90 % of all patterns when using the (8, 1) neighborhood and for around 70 % in the (16, 2) neighborhood. Therefore, little information is lost by assigning all non-uniform patterns to a single arbitrary number. Since only 58 of the 256 possible 8 bit patterns are uniform, this enables significant space savings when building LBP histograms. To indicate the usage of two-transition uniform patterns, the superscript $u2$ is added to the LBP operator notation. Hence the LBP operator with a 2 pixel radius, 8 sampling points and uniform patterns is known as $LBP_{8,2}^{u2}$.

As mentioned previously, LBPs have been used successfully for face description. The LBP feature descriptor is used to build several local descriptors within the facial image. These local descriptors are then combined to yield a global feature descriptor of the face. Feature descriptors build in this way are more robust towards pose and illumination variation.

The basic idea behind constructing LBP feature descriptors can be summarized as follows:

1. The image is converted to gray scale and then divided into smaller number of rectangular blocks

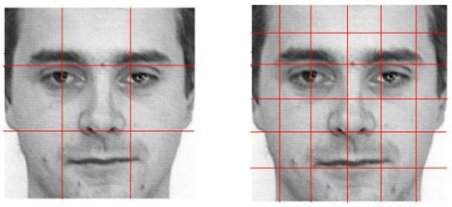


Figure 4.12: Facial Image divided into 3X3 and 6X6 blocks

2. Extract texture descriptors from each region independently. Therefore if the image is divided into m regions, m local histograms are obtained. Let n be the size of each local LBP histogram, and then the combined LBP feature vector has the size $m \times n$. Thus with the spatially enhanced histogram, a description of the image on three different levels of locality is obtained. The LBP labels for the histogram contain information about the patterns on a pixel-level, the labels are summed over a small region to produce information on a regional level and the regional histograms are concatenated to build a global description of the face.

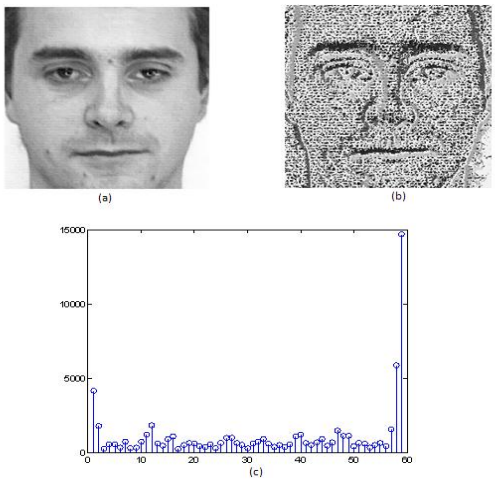


Figure 4.13: (a) Gray scale image (b) LBP coded image (c) $\boldsymbol{LBP}_{8,2}^{\boldsymbol{u2}}$ feature descriptor for (a)

4.6 Feature Extraction Based on Morphological Image Processing

Mathematical morphology [62] is a powerful tool to extract useful feature from images [63] [64]. In this section we describe the process of extracting eyes from facial images using morphological image processing [65]. The basis for eye extraction lies in the fact that eyes are mostly circular in shape and hence image morphology can be applied to the binary edge map resulting in blob like

features on the face. Those blobs can later be classified as eyes depending on certain criteria. Morphological image processing can be binary or grayscale and is heavily based on the concepts of set theory. A binary image can be viewed as a bi-valued function of x and y. Morphological theory views a binary image as the set of its foreground (1-valued pixels) and background (0-valued pixels). Hence set operations like union and intersection can be directly applied to the binary images. Dilation and erosion are the basic morphological operations. Dilation is the operation that *grows* or *thickens* objects in a binary image. The specific manner and the extent of the thickening are however controlled by the shape of the *structuring element.* Mathematically, dilation of a binary image A by a structuring element B can be represented as,

$$A \oplus B = [z|\left(\hat{B}\right)_z \cap\ A\ \neq\ \emptyset\ ] \tag{4.2}$$

Where $\emptyset$ is the empty set and z represents the translations of B.

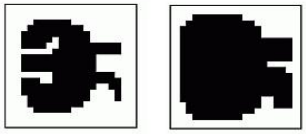


Figure 4.14: Original Binary Image and its dilated version

Morphological erosion is the opposite process of dilation. Erosion essentially *shrinks* or *thins* objects in a binary image. Mathematically the operation of erosion can be represented as

$$A \ominus B = [z|(B)_z \cap\ A^c\ \neq\ \emptyset\ ] \tag{4.3}$$

Hence the erosion of A by a structuring element B is the set of all structuring element origin locations where the translated B has no overlap with the background of A.

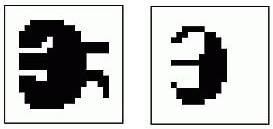


Figure 4.15: Original Binary Image and its eroded version

In many image processing applications the fundamental morphological operations of dilation and erosion are often combined together. The process of "opening" an image will smooth the edges, break narrow block connectors and remove small protrusions. "Closing" an image will also smooth edges but will fuse narrow blocks and fill in holes. The morphological operation of opening is simply erosion of A by B followed by dilation of the result by B. Mathematically opening can be represented as,

$$A \circ B = (A \ominus B) \oplus B \tag{4.4}$$

The reverse of this process is morphological closing. The morphological operation of closing is dilation of A by B followed by erosion of the result by B. The process can be represented mathematically as,

$$A \bullet B = (A \oplus B) \ominus B \tag{4.5}$$

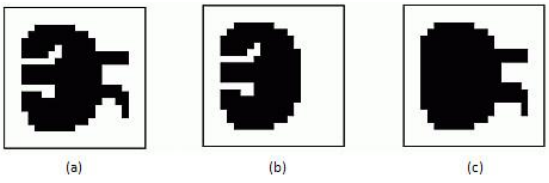


Figure 4.16: (a) Original Binary Image (b) Result after opening (c) Result after closing

A combination of various morphological operations can be used to extract *blob like features*. There are essentially two broad steps in extracting eyes from facial images using morphology:

(1) Extraction of blob like features (2) Classification of the blobs as eyes. Image is first converted to grayscale followed by HVS based edge detection. The result is a binary edge map which is then subjected to a series of morphological operations after which the image essentially consists of several blobs. Morphological dilation is used to enhance the eyes region edges. The image is dilated twice to get the holes in eyes filled. Disk shape structuring element is used in this case due to the circular nature of the eyes. After this step, eyes become filled regions. However, sometime small holes are left inside the eye region which can be filled using morphological hole filling. During the dilation phase, some unwanted edges are also enhanced. To remove these unwanted edges, the image is eroded three times. The classification step then classifies these blobs as eyes based on certain criteria described in Chapter 6.

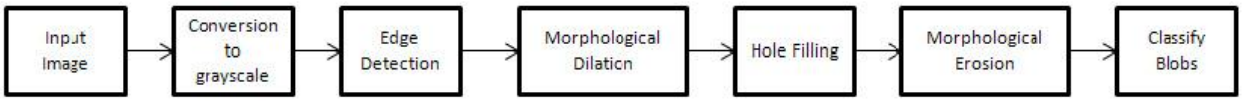


Figure 4.17: Schematic diagram for blob extraction algorithm

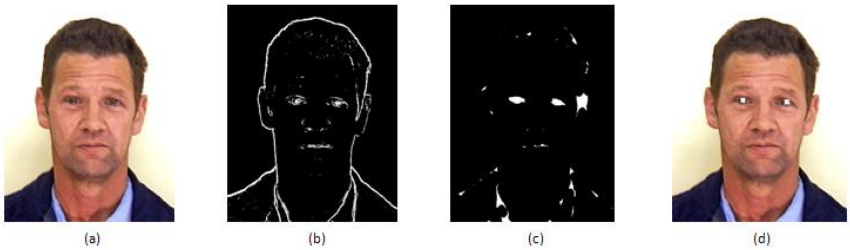


Figure 4.18.1: (a) Original Image (b) Edge Map by HVS (c) Image after morphological processing (d) Classification of blobs

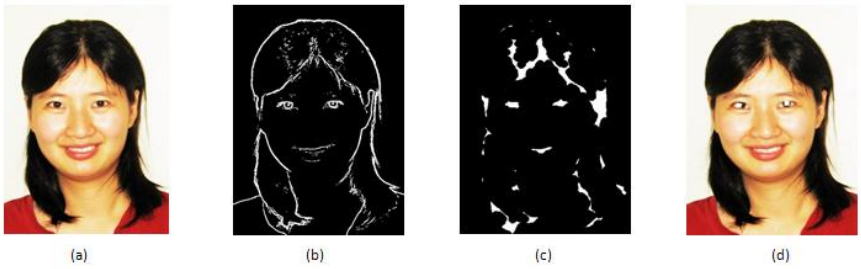


Figure 4.18.2: (a) Original Image (b) Edge Map by HVS (c) Image after morphological processing (d) Classification of blobs

Binary morphology is based on the binary edge map of the input image. Hence we'll analyze the effect of the use of different edge detectors in the process. Specifically we'll analyze the results using the Canny, Sobel and HVS base edge detector. The canny edge detector uses two thresholds to do the final edge labeling on the gradient image. This brings out more details in the image. However the details may not be compatible with the human perception of edges. These are more related to the texture on the image. Another difficulty arises when we apply morphological operations on a canny edge map. Due to dilation, the detailed regions get connected and it is difficult to find distinct blobs representative of the eye. Two eyes get connected resulting in one big connected component instead of two distinct eyes corresponding to the two eyes. Thus the distinct features of the eyes are lost in this case. The results using default thresholds for the canny edge detector are shown in Figure 4.19. Adjusting the threshold may improve the results for one particular image. However it is very difficult to choose a single best threshold that will work for all the images in the database.

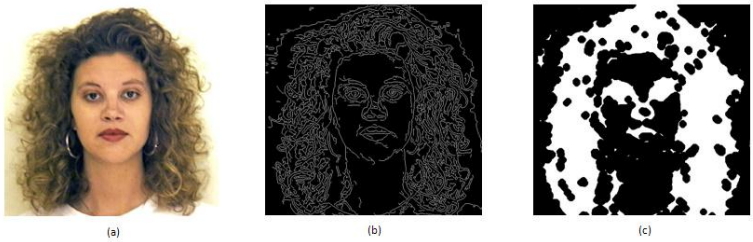


Figure 4.19: (a) Original Image (b) Edge map using the canny edge detector (c) Image after morphological operations

Figure 4.20: Comparison of results using the Canny edge detector with custom threshold values (Low Threshold = .08; High Threshold = 0.2)

The results are better using the sobel edge detector and the HVS based edge detector. The default sobel edge detector in this case outperforms the canny edge detector. The results using the sobel edge detector is shown in Figure 4.21. The same set of images is preprocessed using an HVS edge detector and the results are shown in Figure 4.22.

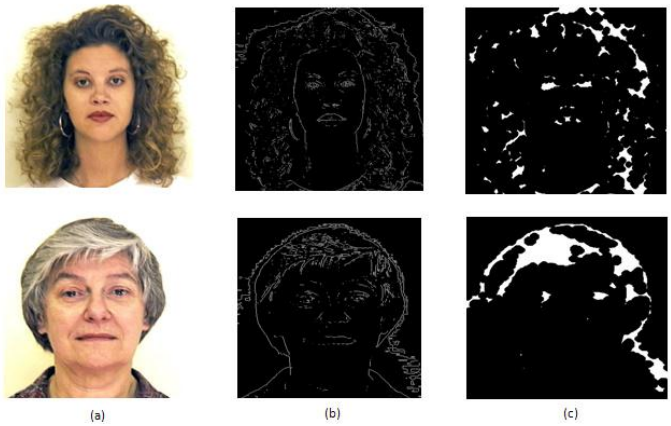


Figure 4.21: (a) Original Images (b) Edge maps using the default sobel edge detector (c) Images after Morphological Operations

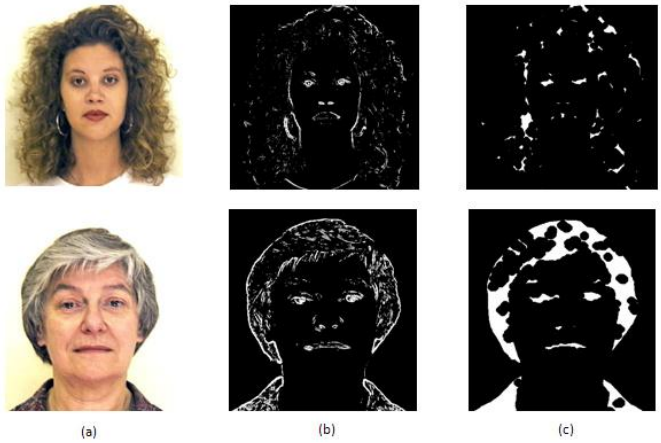


Figure 4.22: (a) Original Images (b) Edge maps using the HVS edge detector (c) Images after Morphological Operations

4.7 Feature Extraction by Principal Component Analysis (PCA)

The Principal Component Analysis (PCA) is a useful statistical technique that has been widely used in applications like face recognition, image compression and pattern recognition in high dimensionality data [66] [67] [68] [69] [70]. PCA is a way of identifying patterns in data and

expressing the data in a way as to highlight their similarities and differences. Once the patterns within the data have been recognized, the task of data compression becomes easier without the loss of significant information. Hence PCA is widely used for image compression. In this thesis, PCA is used as a technique for face recognition from images. The main idea of using PCA for face recognition is to express the large 1-D vector of pixels constructed from 2-D facial image into the compact principal components of the feature space. This is called eigenspace projection. Eigenspace is calculated by identifying the eigenvectors of the covariance matrix derived from a set of facial images. A 2-D facial image can be represented as 1-D vector by concatenating each row (or column) into a long thin vector. A 2-D facial image can be represented as 1-D vector by concatenating each row (or column) into a long thin vector. Let us assume that we have M vectors of size N (= rows of image X columns of image) representing a set of images. If $p_j's$ represent the pixel values then the vector x_i can be represented as,

$$x_i = [p_1 \ldots p_N]^T, i = 1, \ldots, M \tag{4.7}$$

The images are mean centered by subtracting the mean image from each image vector. Let m represent the mean image.

$$m = \frac{1}{M} \sum_{i=1}^{M} x_i \tag{4.8}$$

Let w_i represent the mean centered image, then

$$w_i = x_i - m \tag{4.9}$$

The objective of PCA based feature extraction is to find a set of $e_i's$ which have the largest possible projections onto each of the $w_i's$. In other words, our aim is to find a set of M orthonormal vectors e_i for which the following quantity

$$\lambda_i = \frac{1}{M} \sum_{n=1}^{M} (e_i^T w_n)^2 \tag{4.10}$$

is maximized with the orthonormality constraint

$$e_l^T e_k = \delta_{lk} \tag{4.11}$$

Now $e_i's$ and $\lambda_i's$ are given by the eigenvectors and eigenvalues of the covariance matrix, C.

$$C = WW^T \tag{4.12}$$

Where W is the matrix formed by placing the column vectors $w_i's$ side by side. The size of C in this case is NXN, which is enormous. Hence it is not practical to solve for the eigenvectors and eigenvalues of C directly. From the principles of linear algebra we know that the vectors $e_i's$ and the scalar $\lambda_i's$ can be obtained by solving for the eigenvectors and eigenvalues of the MXM matrix W^TW. Let d_i and μ_i be the eigenvectors and eigenvalues of W^TW respectively, then we have

$$W^TWd_i = \mu_i d_i \tag{4.13}$$

Multiplying both sides by W, we have

$$WW^T(Wd_i) = \mu_i(Wd_i) \tag{4.14}$$

Hence the first M eigenvectors e_i and eigenvalues λ_i of WW^T are (Wd_i) and μ_i respectively.

The eigenvectors corresponding to nonzero eigenvalues of the covariance matrix produce an orthonormal basis for the subspace within which most image data can be represented with a small amount of error. The eigenvectors are sorted from high to low according to their corresponding eigenvalues. The eigenvector associated with the largest eigenvalue is one that

reflects the greatest variance in the image and the smallest eigenvalue is associated with the eigenvector that finds the least variance. They decrease in exponential fashion, meaning that the roughly 90% of the total variance is contained in the first 5% to 10% of the dimensions. The projection of facial image onto $M'(< M)$ dimensions is thus,

$$\Omega = [v_1 v_2 \dots v_{M'}]^T \tag{4.15}$$

Where $v_i = e_i^T w_i$, and v_i is the i^{th} coordinate of the facial image in the new space obtained by PCA. The vectors $e_i's$ are also images and hence they are called eigenimages or eigenfaces and they represent the feature space onto which the training and test images can be projected.

The face database used for our experiments is obtained from AT&T Laboratories, Cambridge. It contains ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). Figure 4.23 shows five of the training subjects.

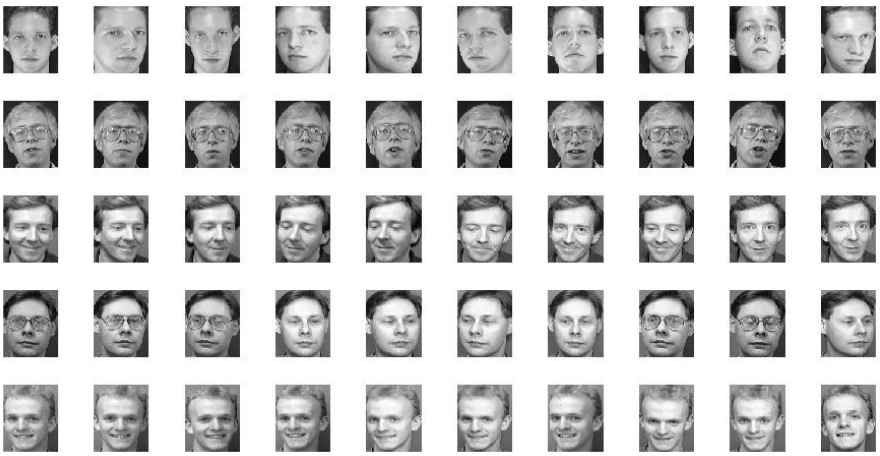


Figure 4.23: Training images from AT&T Laboratories, Cambridge

Figure 4.24: Mean facial image

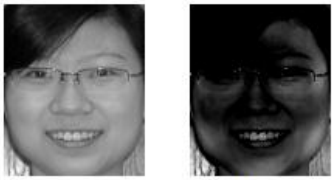


Figure 4.25: (a) Grayscale Image (b) Mean Subtracted Image

Figure 4.24: Top 20 Eigenfaces from the database

CHAPTER 5

HUMAN VISUAL SYSTEM BASED OBJECT DETECTION AND RECOGNITION

A typical object detection and recognition application in general has three phases: namely, training, testing and classification. In the training phase, depending on the application, a training database is constructed containing images of objects that are of interest in the application. Usually along with the objects of interest, a separate set of images are also considered which consists of objects that are not of interest in the application however which can be present in the images along with the object of interest. For example is we consider a detection system which is used to detect pair of eyes from facial images, then eyes are categorized as the objects of interest in the application. A facial image also has other facial parts like nose; lips; ears etc. and these objects are not of interest in the application. A feature vector generation method is then applied to the training database for extracting feature vectors from the training images. Sometimes more than one feature extraction technique are employed and the feature vectors extracted from each step are then fused together to get a more robust representation of the object.

In the testing phase, the test image is gathered using image acquisition techniques. Depending on the application and method of acquisition, the image may be subjected to preprocessing activities like noise suppression or contrast enhancement. Feature vectors are then extracted from the test image and compared with the feature vectors obtained in the training images. The method for extracting feature vectors needs to be consistent throughout the process. In other words the same set of feature vectors should be used in the training and the actual testing phases.

In the classification phase, the feature vectors obtained from the test image are compared with the feature vectors extracted in the training database and classified according to their similarity

with feature vectors from the training database. The schematic diagram of a typical object detection system is shown in Figure 5.1.

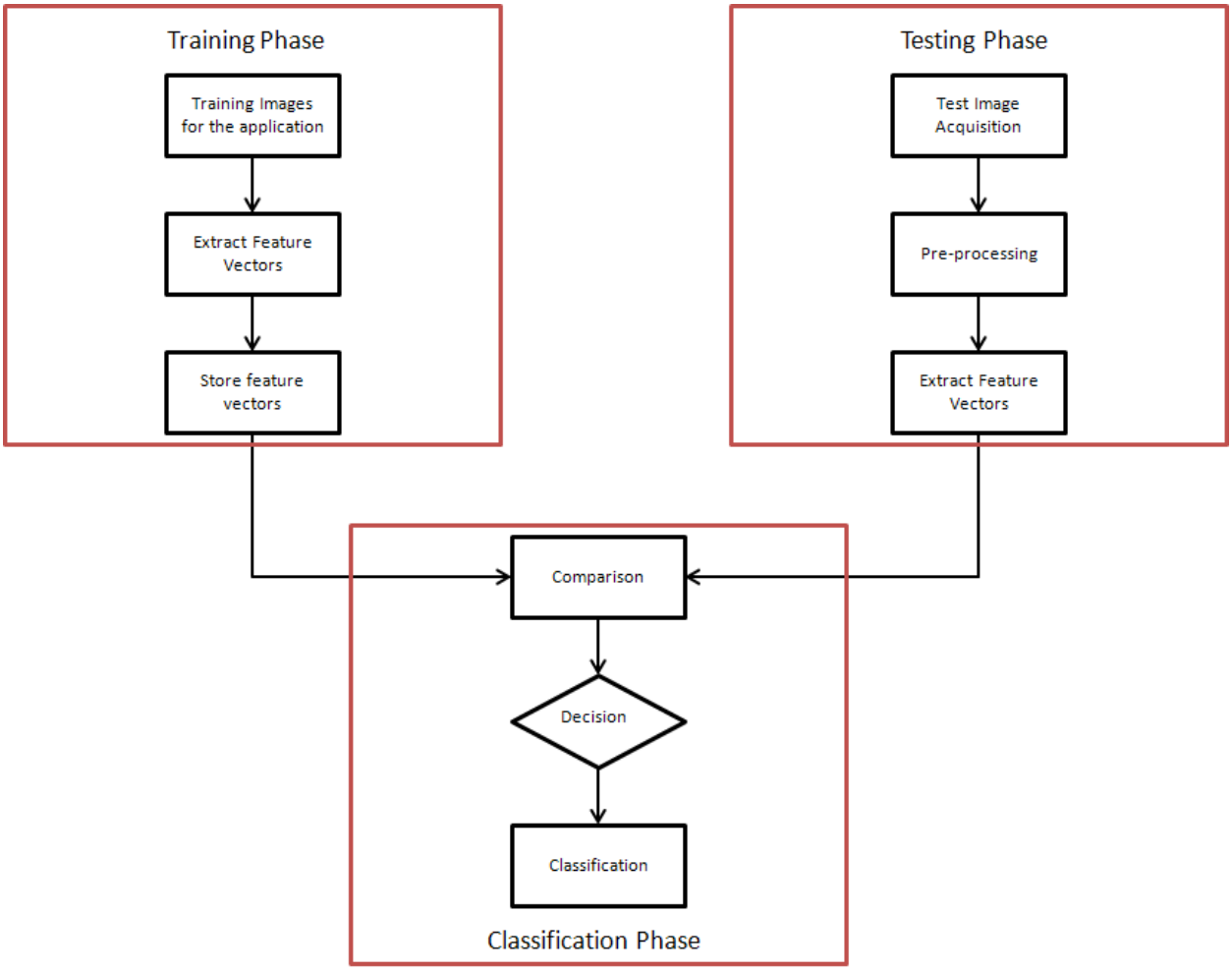


Figure 5.1: Schematic Diagram of a typical Object Detection System

5.1 Generic Framework for HVS based Object Detection and Recognition

In this thesis we propose a novel architecture for human visual system based object detection and recognition. Human visual system based image decomposition is incorporated into the different phases of the typical object detection and recognition system. Figure 5.2 shows the

schematic diagram for HVS based object detection and recognition system. The blocks bordered in green shows the additional HVS based processing incorporated in the generic architecture.

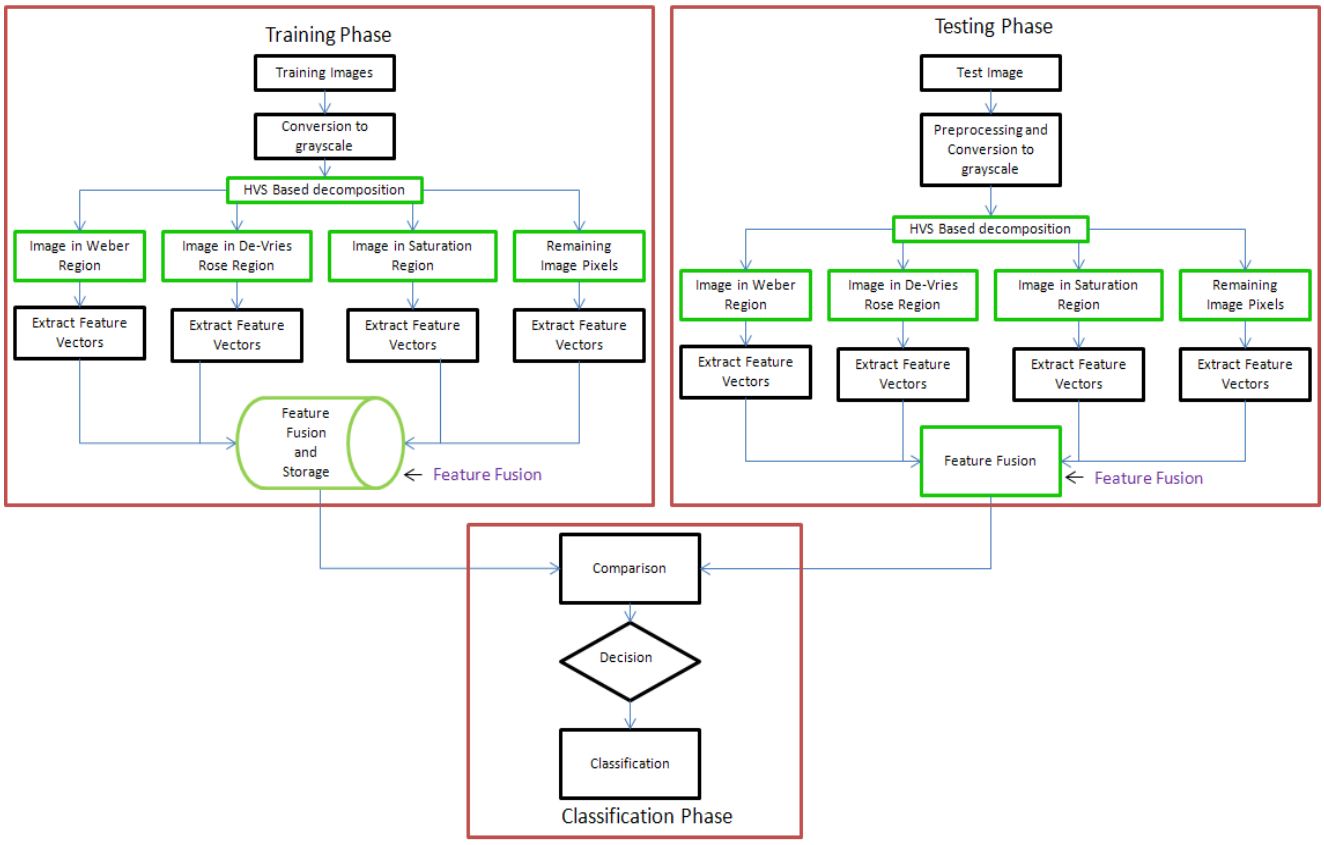


Figure 5.2: Schematic Diagram of Object Detection System incorporating HVS based image decomposition

In the training phase, we decompose the training images into HVS based sub-images and extract feature vectors from each of them. The feature vectors thus obtained are weighed and are combined together, a process referred to as feature fusion. Feature fusion can be done using arithmetic, logical or even logarithmic addition. In our experiments for detecting eyes from facial images, we have used arithmetic and logical operations for feature fusion and have

obtained comparable results. Feature fusion can also be accomplished using concatenation of the features obtained from the different HVS based sub-regions.

In testing phase the test image undergoes HVS based decomposition and feature extraction from each of the sub-images followed by the feature fusion step. From our experiments we have found that majority of the information about images taken under good lighting conditions, tend to be concentrated in the Weber region of HVS. Hence we often combine the De-Vries Rose and the saturation regions with the Weber region and then extract feature vectors from the combined region. Also instead of extracting features from the image containing the "remaining image pixels", the original image can be used.

The classification phase classifies the feature vectors obtained from the test images by comparing them with feature vectors obtained from the training images. Classification can be based on a set of rules applied to the feature vectors obtained in the test image. This method has been used in the eye detection system described in Chapter 6. In more generic terms classification consists of comparing feature vectors from the training and testing database and then taking a decision depending on the similarity of the training and testing feature vectors. The similarity is usually found by using a classifier. The classifier used in this type of classification can be nearest-neighbor classifiers using Manhattan or Euclidean distances or can be more sophisticated form of classifiers like the Support Vector Machine (SVM) or Neural Networks. In our experiments for modified LBP based face recognition discussed in chapter 7, we have used nearest-neighbor classification approach.

CHAPTER 6

HUMAN VISUAL SYSTEM BASED EYE DETECTION

Detection of eyes from facial images is a crucial aspect in many image processing applications like face recognition/detection, driver behavior analysis or gaze estimation for understanding human behavior. It has been found that majority of traffic accidents are caused due to driver's drowsiness [71]. Hence automated systems are now developed that can track the eye of the driver in subsequent frames to detect the state of the eye (open/close) and warn the driver in case the eyes remain closed for a considerable amount of time. Another important application involves gaze estimation for security and surveillance purposes.

Several methods for eye detection have been proposed. In majority of them, face detection is performed as a preprocessing step prior to detecting the eyes. It has been observed that eyes possess strong horizontal edges. Hence often directional image vectors are projected horizontally to detect the local minima which correspond to the y-coordinates of the eyes [72]. Also the pupils are generally darker than the surrounding facial region. Some facial feature extraction algorithm searches for local gray minima within segmented facial regions [73]. Applications have been developed which takes into account the circular structure of the eye. Artificial circular templates are often built and the correlation coefficient is calculated between the template and the eye image[74] [75]. Hough transform has also been used for eye detection since it can detect shapes like circle or ellipse [76].

In this chapter we present an algorithm for eye detection which is based on extracting blobs from facial image using morphological image processing and then classifying the blobs as eyes.

This human visual system based algorithm is based on the generic framework for object detection introduced in Chapter 5.

6.1 Flow Diagram for Human Visual System Based Eye Detection

The flow diagram of the human visual system based eye detection is presented in Figure 6.1. The blocks bordered in green shows the additional processing steps for incorporating HVS based image decomposition.

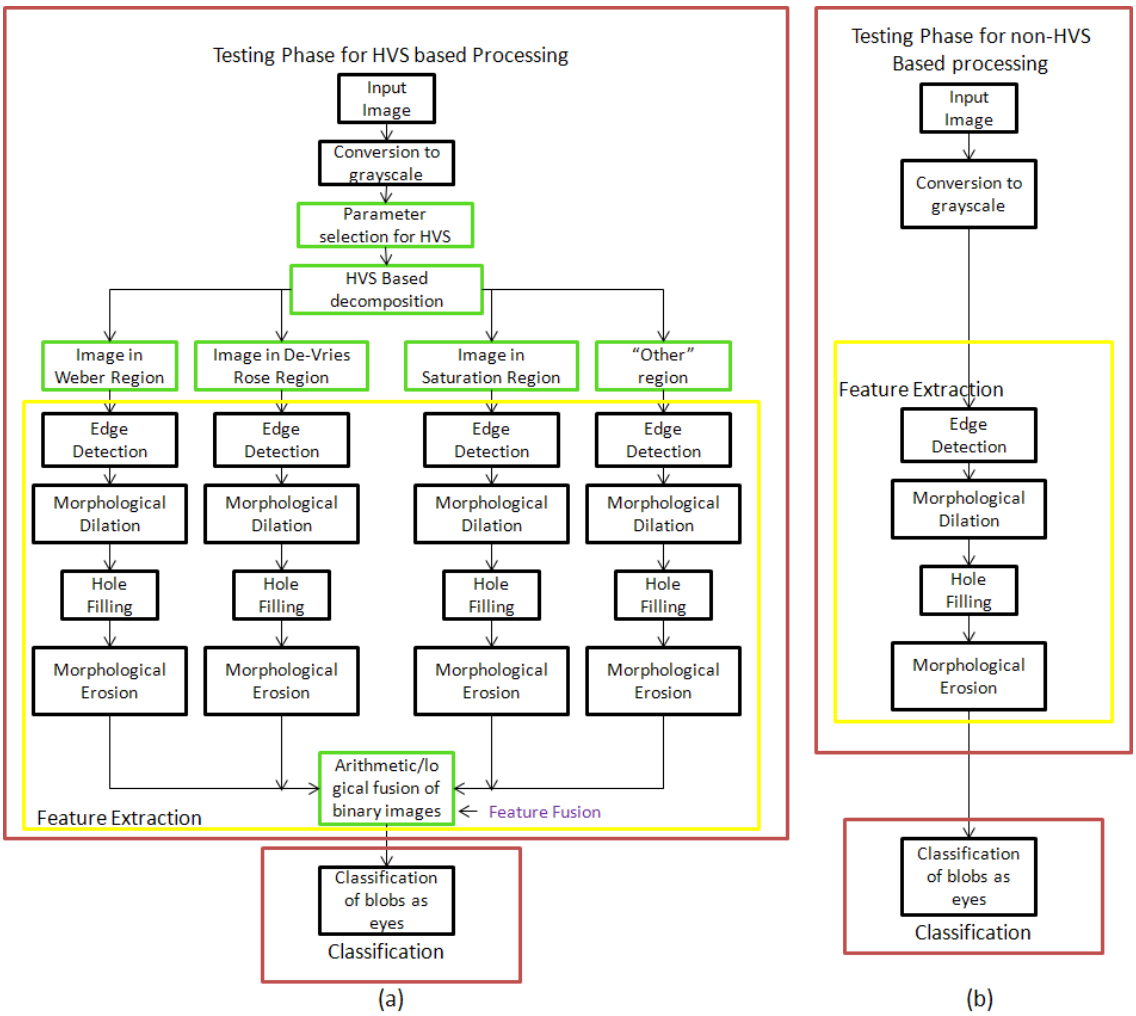


Figure 6.1: Schematic Diagram for Eye detection Based on (a) HVS based Image Decomposition and Morphological Image Processing (b) non-HVS based processing

Eyes are generally represented as circular structures and hence the objective of eye detection algorithm is to look for circular features in the facial image. Morphological operations are usually performed on binary images. In this algorithm, the decomposed images in the different regions are converted to binary images by applying the Sobel edge detector. Since the algorithm processes binary images after the edge detection step, this algorithm is computationally efficient and is suited applications where hardware utilization is a constraint. Further, the training database is not required in this case because eyes are represented as circular structures on the binary image. Hence the objective is to find blob like features from the image after morphological operations.

The algorithm takes in an input image and converts it to grayscale. An example input image and its grayscale version is shown in Fig. 6.2(a). It is assumed that the input image satisfies the flowing conditions:

- The image is a head-shoulder image.
- Both eyes are visible.
- The head is tilted by not more than 45 degrees
- The eyes are not closed

The different steps involved in the eye detection algorithm are described below:

6.1.1 Image Decomposition

The 2-dimensional image is first subjected to HVS based decomposition resulting in four images; in the Weber region, De-Vries Rose region, Saturation region and the "Other" region. The image decomposition in accordance with the human visual response has been discussed in Chapter 3 of this thesis.

6.1.2 Edge Detection

Binary morphology has been used in the algorithm to extract blob like features from the facial images. Each of the images is subjected to edge detection in the next step. In our experiments the Sobel edge detector has been used. The canny edge detector tends to bring out more details in the image using 2 threshold values. When we apply morphological operations on a canny edge map, due to dilation, the detailed regions get connected and it is difficult to find distinct blobs representative of the eye.

6.1.3 Morphological Operations

The binary edge maps are subjected to morphological operations namely dilation, hole filling and erosion. Dilation grows or thickens objects in a binary image. Hence morphological dilation has been used in this case to enhance the eyes region edges. The image is dilated twice using circular structuring elements of radius 4. It would be shown later that there is a dependency of the size of the structuring element used on the size of the image. If the average size of the facial images in the database is around 450 X 350, then using a structuring element of radius 4 gives best results. The edges near the eyes thicken and holes in the eyes get filled after this step. To make sure no holes remain in the eye region, an additional step of hole filling is incorporated. During the dilation phase, some unwanted edges are also enhanced. To remove these unwanted edges, the image is eroded three times. Fig 6.2(b), 6.2(c) and 6.2(d) shows the results of morphological dilation and erosion in the Weber, “Other” and Saturation region. The De-Vries Rose region does not contain any useful information and hence not shown.

6.1.4 Fusion

The morphologically processed images in the different regions of the human visual response are then fused using arithmetic operations. Logical addition can also be used in this case since the images after morphological operations are binary images. From our experiments we have seen that for well illuminated images, majority of the information lies in the Weber and Other region of the image. Hence the De-Vries Rose and the Saturation regions can be excluded from processing thereby reducing the computational overhead. However for images that have non-uniform illumination and regions of shadows around the eyes, we need to use all the components. It has been found from our experiments that HVS based image decomposition can be used to extract more information from images having shadows by increasing the span of the De-Vries Rose region. In images having non-uniform illumination, the region near the eyes tends to get darker. The rate of increase of threshold in the De-Vries Rose region is slower compared to the Weber region and hence this region along with the Weber and "Other" is used to extract blobs from images having non-uniform illumination. After morphological operations, the resultant binary image contains blobs which need to be classified as eyes. Fig 6.2(e) (left) shows the result after fusion.

6.1.5 Classification

In the classification step, the blobs detected are classified as eyes. This is based on the following set of rules applied sequentially to the binary image obtained after morphological operations:

- The aspect ratios (width/height) of the blobs corresponding to the eyes regions are between upper and lower thresholds. The typical thresholds chosen for our experiments are 0.8 and 8.

- The orientation angle of eyes is not greater than 45 degrees. This can be done by choosing a threshold to restrict the vertical and horizontal positions of the eyes.
- Size of both eyes is comparable. Here we assume that the ratio of the size of the blobs corresponding to the eyes should be between 0.4 and 2.5.
- Depending on the size of the image, the distance between the two eyes should lie between a minimum and maximum threshold value.
- Also if we assume that the input is a facial image, an additional condition can be imposed which states that the eyes should not be close to the border of the image. This condition reduces the rate of false positives in detection.

The final result is shown in Fig 6.2(e) (right).

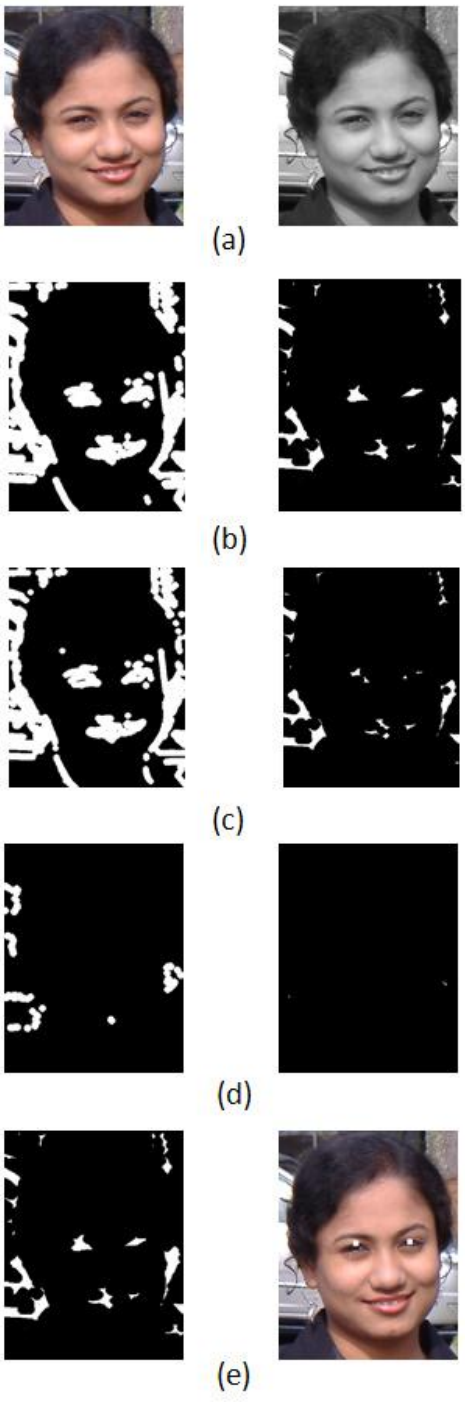


Figure 6.2: (a) Original Image and its grayscale version (b) Result after dilation and erosion in the Weber region (c) Result after dilation and erosion in the "Other" region (d) Result after dilation and erosion in the Saturation region, (e) Image after fusion (left) and Final Result (right)

6.2 Results and Discussion

6.2.1 Tests using good images

The algorithm is tested using the MATLAB Image Processing Toolbox. The algorithm has been tested using PICS facial database (PICS, 2003: Psychological Image Collection at Stirling, available from <http://pics.psych.stir.ac.uk/>, University of Stirling Psychology Department). 92 images were selected from the database where the images were taken under regular illumination condition. In addition to the images from the database, we have also used test images of the members of SIMLAB. The facial images are mostly frontal facial images and some of the subjects had glasses on. In addition to the well-illuminated images, we have taken 70 pictures from the database where the images had non-uniform illumination and shadows around the eye region. As mentioned previously, the regions surrounding the eyes tend to get darker when the images have varying illumination. The De-Vries Rose region of HVS has been extended for testing these images and results have been compared with non-HVS processing.

Figure 6.3 shows some of the results using images having good illumination. Table 6.1 and 6.2 show the percentage accuracy and compare the HVS based method with the original non HVS based algorithm. We compare at two stages of the algorithm: the blob like feature extraction stage and the final eye extraction stage after classification of eyes.

The alpha and beta parameters for HVS decomposition have been determined experimentally. For images having good illumination, the optimum parameter values are $\alpha_1 = 0$; $\alpha_2 = 0.1$; $\alpha_3 = 0.9$; $\beta = 0.05$. We have also showed the variation of the feature extraction rate (after blob extraction and before classification) with the parameter beta which is the thresholding parameter for HVS decomposition. The plot is given in Figure 6.4.

Figure 6.3: Experimental Results using well-illuminated images

METHOD	TOTAL IMAGES	CORRRECT DETECTION	PERCENTAGE ACCURACY
HVS Based	92	86	93.4
Non-HVS	92	74	80.4

Table 6.1: Comparison of Detection Rate in Feature Extraction Step

METHOD	TOTAL IMAGES	CORRRECT DETECTION	PERCENTAGE ACCURACY	NUMBER OF FALSE POSITIVES
HVS Based	92	80	86.9	6
Non-HVS	92	68	73.9	7

Table 6.2: Comparison of Detection Rate in the final Eye Detection Step

The false positives are reduced by applying an additional constraint that there is only one pair of eye in the facial image. If we remove this constraint, the number of false positives increase to 19 for HVS based processing and 15 for non-HVS based processing.

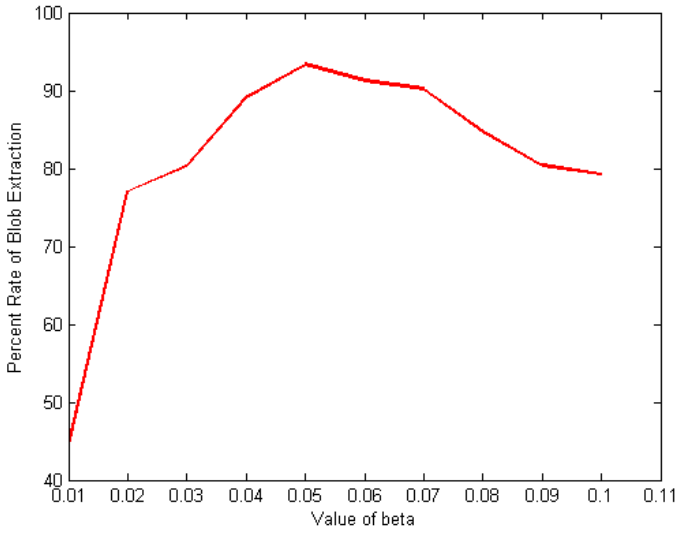


Figure 6.4: Variation of the rate of change of blob extraction rate vs. the HVS thresholding parameter Beta: The results outperforms non-HVS based processing for a range of beta from .03-.09

6.2.2 Tests using Images having Non-Uniform Illumination and shadows around Eye regions

We used 70 images having non-uniform illumination from the same database. In this case we increased the span of the De-Vries Rose region of HVS by alpha-parameter variation.

In images having non-uniform illumination, the region surrounding the eyes tends to have shadows. The De-Vries Rose region has a lower threshold compared to the Weber and Saturation regions of HVS and hence this region tends to bring out information from regions having darker backgrounds. HVS based image decomposition also helps in detection of eyes from images that have varying illumination across different parts of the image. While the De-Vries Rose region has more information about the darker background regions, the well-illuminated regions are still captured in the Weber and Other regions. Figure 6.5 shows an example. Table 6.3 compares the rates of detection using HVS based and non-HVS based processing on images with non-uniform illumination. Figure 6.6 and 6.7 show experimental results using HVS and non-HVS based processing.

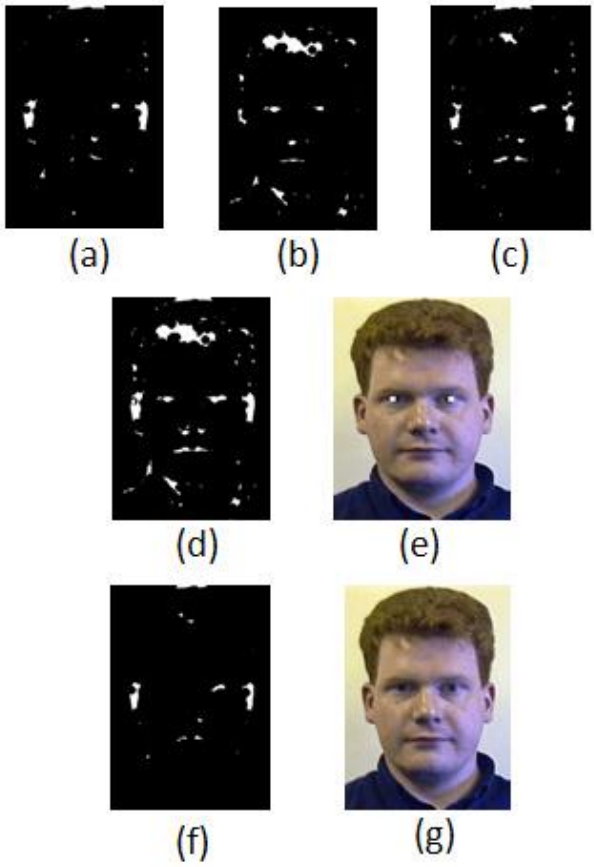


Figure 6.5: (a) Results after morphological operations in the Weber Region, (b) Results after morphological operations in the De-Vries Rose Region, (c) Results after morphological operations in the "Other" Region, (d) Fusion of (a), (b) and (c), (d) Final Result (e) Results after morphological processing with non-HVS method, (f) Final result without HVS decomposition

The optimum values of the HVS parameters used for testing these images are: $\alpha_1 = 0$; $\alpha_2 = 0.3$; $\alpha_3 = 0.9$; $\beta = 0.03$

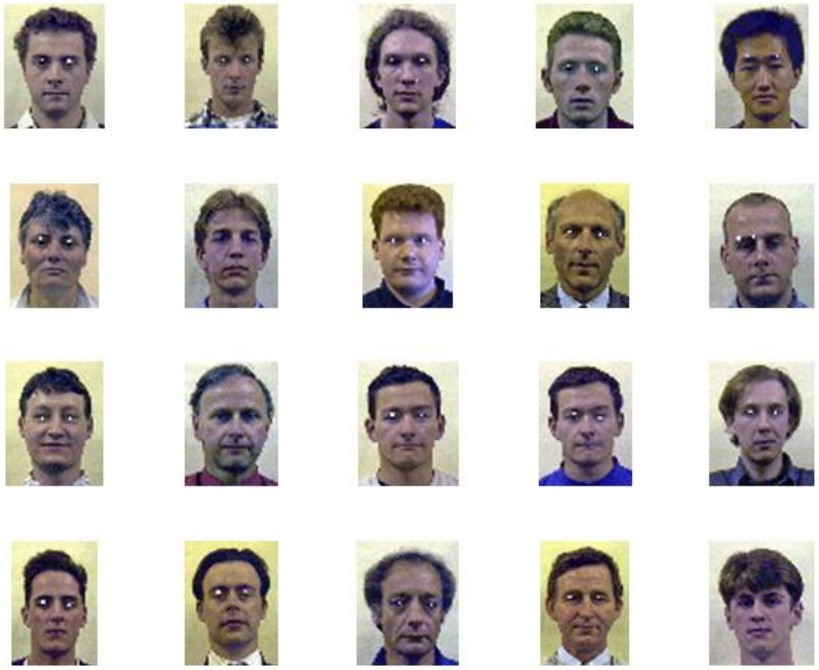


Figure 6.6: Experimental Results using HVS based processing on images having Non-Uniform Illumination and shadows around the eye regions

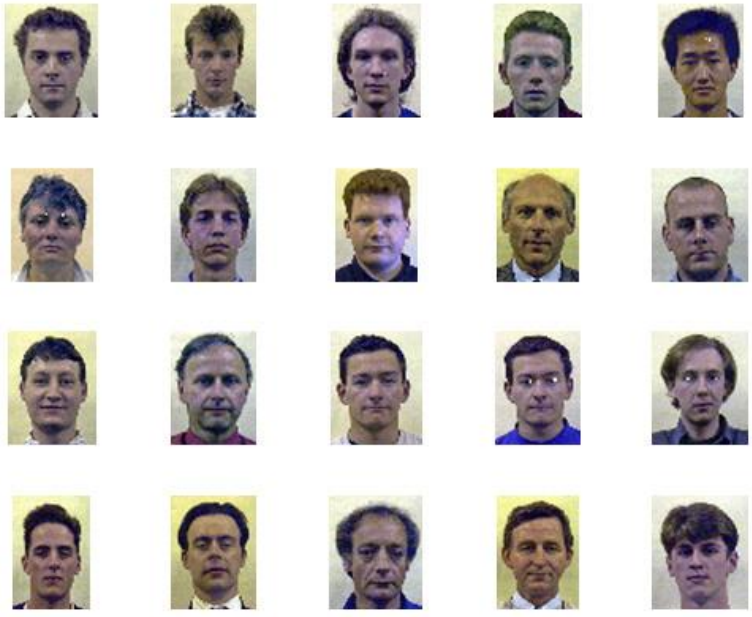


Figure 6.7: Experimental Results using Non-HVS based processing on images having Non-Uniform Illumination and shadows around the eye regions

METHOD	TOTAL IMAGES	CORRRECT DETECTION	PERCENTAGE ACCURACY
HVS Based	70	37	53
Non-HVS	70	14	20

Table 6.3: Comparison of Detection Rate in the final Eye Detection Step using Images having Non-Uniform Illumination and shadows around the eye regions

6.2.3 Tests on an Image after introducing Shadows and Tint

We have manually introduced shadows and tint in a facial image before eye detection. Figure 6.8 shows the results of HVS based algorithm and compares the performance with the non-HVS based algorithm. Under perfect illumination, both of them work very well. However under differential lighting conditions, the non-HVS based processing fails to detect the pair of eyes.

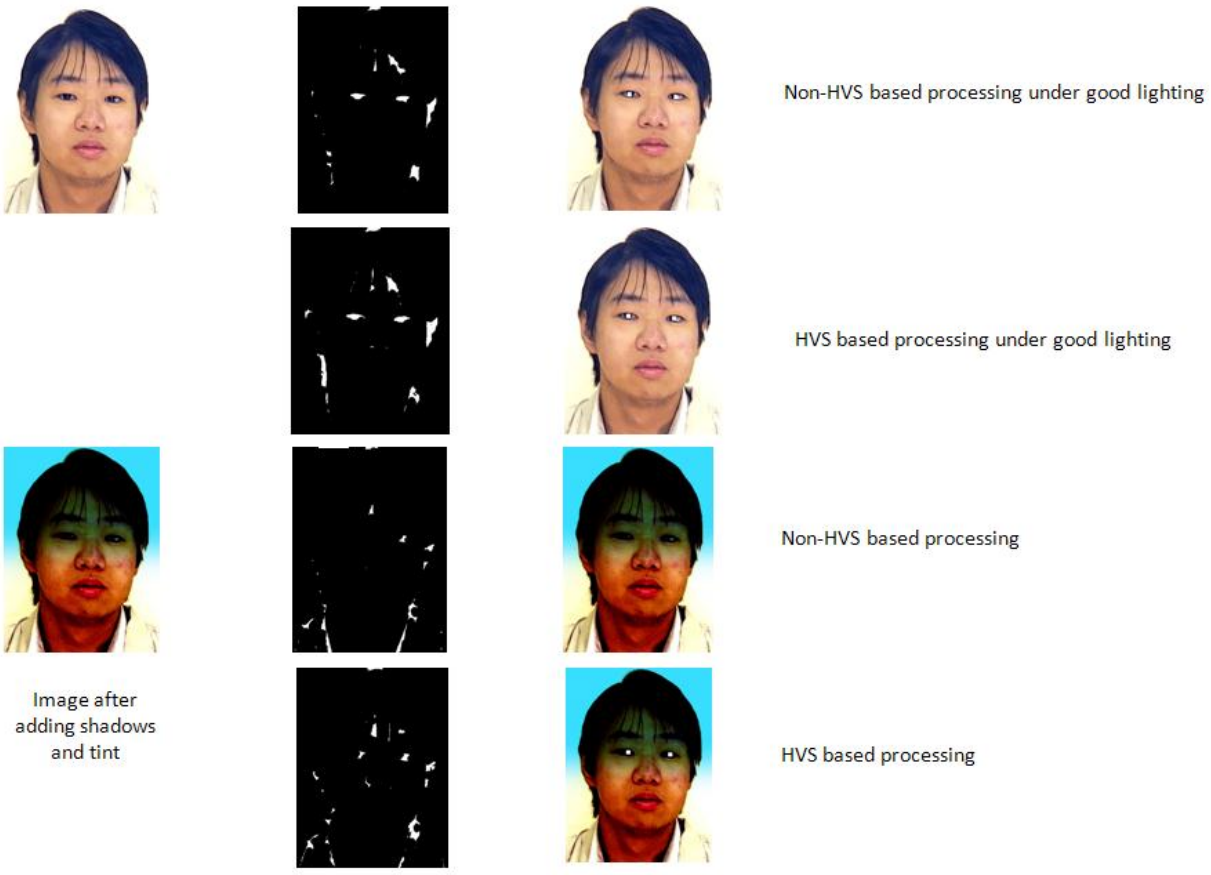


Figure 6.8: Test on image after introducing shadows and tint: HVS based processing detects eyes successfully

6.2.4 Tests on an Image where Subjects have Glasses on

The feature extraction algorithm does not function properly when subjects have their glasses on.

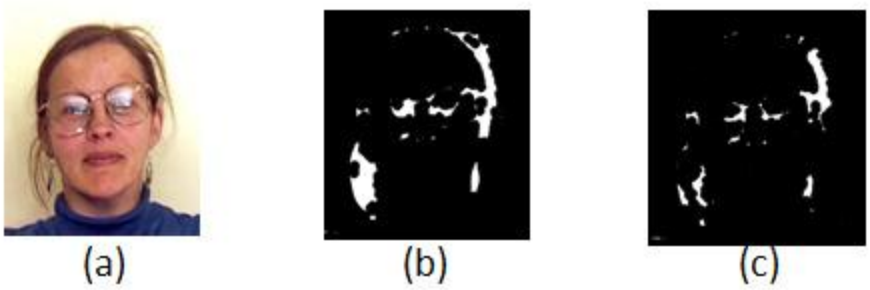


Figure 6.8: (a) Original Image (b) Feature extraction after non-HVS based processing (c) Feature Extraction after HVS based processing

Both HVS and non-HVS based processing fail to extract blob-like features corresponding to the eyes in presence of glasses.

6.2.5 Tests on Image having Multiple Faces

Figure 6.9: (a) Original Image having 2 faces (b) Results after feature extraction through HVS based processing (c) Final Result after HVS based processing (d) Final Result after non-HVS based processing

Figure 6.9 shows an image having 2 faces on a more or less uniform background. Both HVS and non-HVS based processing works well under such circumstances. However if there are multiple faces on a background that is non-uniform (for example images taken outdoor) face detection needs to be added as a preprocessing step for both HVS and non-HVS based processing.

6.2.6 Tests on Image taken with Cell Phone Camera

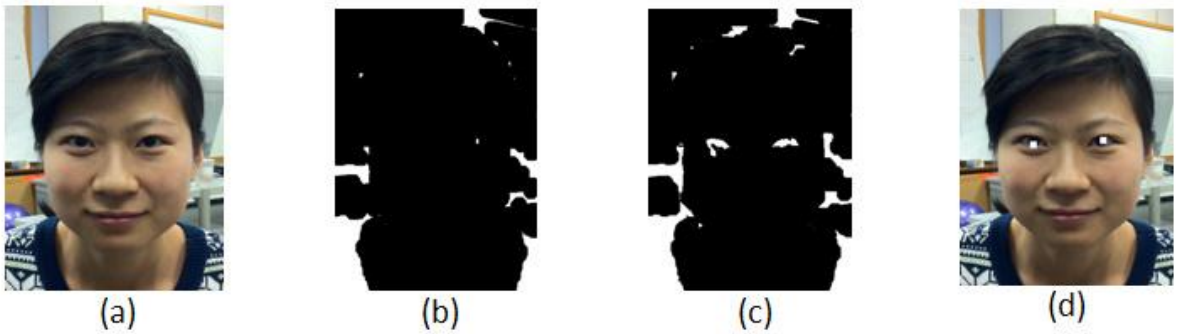


Figure 6.10: (a) Original image taken with cell phone camera (b) Feature extraction using non-HVS based processing (c) Feature Extraction by HVS based processing (d) Final Result using HVS based processing

Figure 6.10 shows an image taken with cell phone camera and shows the final result using HVS based processing.

6.2.7 Effect of the Size of the Structuring Element

Morphological image processing operations are performed by using a structuring element. A structuring element is basically a predefined shape which is used to probe an image drawing conclusions on how this shape fits or misses the shapes in the image. In this case the shapes in the image that are of interest to us are circular blob like features. Hence the size of the structuring element needs to be chosen properly so that after morphological operations, only the blobs corresponding to the eyes remain. The database of images used for our experiments have average size of 450 X 350. Disk shaped structuring elements of size 4 gives optimum

performance in this case. If the input test images have sizes that largely differ from the average size, blobs corresponding to the eyes may not be detected. Hence input images are resized.

Figure 6.11: (a) Original Image having size 78 X 78 (b) Morphological operations on (a) with structuring element having size 4 (c) Morphological operations on (a) after resizing image to size 450 X 350 (d) Original Image having size 2592 X 1936 (e) Morphological operations on (d) with structuring element having size 4 (f) Morphological operations on (d) after resizing image to size 450 X 350

6.3 Execution Time and Memory Usage

In this section we compare the execution time and the memory usage of the HVS vs. the non-HVS based approach. The following table compares the time of execution for eye detection using 92 test images

Processing Time HVS	Processing Time non-HVS
395 seconds	100 seconds

Table 6.4: Comparison of the execution time for HVS vs. non-HVS approach

Thus we see that using the HVS-based system, the processing time required for detecting eyes from 92 test images is about 4 times than using the non-HVS based approach. This happens due to the fact that, in the HVS based approach four images are used in place of one in the original approach. This also implies that the CPU and memory usage will be 4 times than while using the original non-HVS based approach. For example, if the non-HVS based approach requires "N" ALU operations (addition, subtraction etc.), the non-HVS based approach would require "4N" ALU operations. Hence we see that there is a tradeoff involved in using the HVS based approach against the original approach. However since the four images from HVS based image decomposition can be processed independent of one another, distributed systems can be use and parallel processing can be leveraged.

6.4 Limitations

In this section we present the limitations of the eye detection algorithm. We found that when the subjects have their glasses on, both the HVS and the non-HVS based approaches fail to detect the features using morphological image processing. Another difficulty arises when there are multiple faces present in the image. We have shown examples where we have two facial images against a more or less uniform background and seen that using this method we can detect the eyes. However when there are multiple faces present against a heterogeneous background (for example when images are taken against trees, houses etc.), morphological feature extraction does not work so well. Hence we propose a face recognition step as a pre-processing step for the algorithm.

We also showed the effect of the size of the structuring element. Hence image resizing is another preprocessing step that needs to be included in this algorithm.

Another limitation arises when using the images that have non-uniform illumination and shadows near the eye region. Though using our approach has resulted in significant improvement in the detection rate, further improvement is possible. We propose a preprocessing step using the different image enhancement algorithms.

6.5 Summary

In this section we summarize the observations and discuss the pros and cons of the HVS based algorithm for eye detection:

- This algorithm runs faster compared to template matching algorithms using sophisticated classifiers like support vector machines or artificial neural networks. Also by classifying eyes as circular blobs, we can exclude the training database thereby reducing system overhead. We have performed experiments using support vector machine classifiers and found that the number of false positives can be huge if the templates representing non-eye regions are not exhaustive enough. It is really hard to reduce the false positive rate such methods.
- The algorithm performs better than the non-HVS based processing on images having non-uniform illumination and shadows around the eye regions.
- The feature extraction based on HVS outperforms the results using non-HVS based processing.

CHAPTER 7

FACE RECOGNITION BASED ON LOGARITHMIC LOCAL BINARY PATTERNS

Automatic face analysis which includes, e.g., face detection, face recognition, and facial expression recognition has become a very active topic in computer vision research and considered as one of the fundamental problems in computer vision [77]. Face recognition has many applicable areas like identification, face classification, or sex determination. The most useful applications contain crowd surveillance, video content indexing, personal identification, entrance security, etc. Face recognition often becomes a difficult problem because of the generally similar shape of faces combined with the numerous variations between images of the same face. A key issue in face analysis is finding efficient descriptors for face appearance. The human visual system can identify faces even under degraded viewing conditions like viewpoint, illumination, facial expression, occlusion, disruption due to accessories and so on [78]. However, automated face recognition is not yet able to achieve comparable results because measuring the similarity between two faces is based on the conventional measures of image similarity, such as, Euclidean metric or Normalized correlation. The key challenges in facial recognition are

1. A large variability in facial appearance of the same person which is caused by variations of facial pose, illumination, and facial expression
2. High dimensionality of data and small sample size which happens due to the lack of reference images to train face templates.

In real-life applications, the enrollment procedure has to be fast and is generally done once. The few available training data are usually not enough to cover the intra-personal variability of the face. Hence a key issue in face recognition is finding efficient descriptors for facial appearance

[79]. Zhao et.al presents an extensive survey of the different approaches for face recognition in [80].

7.1 Face Recognition: Existing Methods

Currently, many holistic approaches have been used as facial feature descriptors. One of the methods that yield promising results on frontal face recognition is the principal component analysis (PCA), where facial images are expressed as a subset of their eigenvectors, and hence called eigenfaces [66]. Another well-known method for face recognition is the Fisherface approach [81] [82]. The Fisherface approach for face recognition is based on linear discriminant analysis (LDA). LDA extracts the most discriminant features of a facial image by using a set of projection vectors to project the sample to form minimum within-class scatter and maximum between-class scatter.

Holistic methods are however sensitive to the variations of pose, illumination etc. Hence local descriptors of faces have gained wide popularity. One of the first face descriptors based on information extracted from local regions is the eigenfeatures method proposed by Pentland et al. [83]. In this approach the features are obtained by performing PCA to local face regions independently. Another approach is the Local Feature Analysis [84] in which kernels of local spatial support are used to extract information about local facial components. Elastic Bunch Graph Matching (EBGM) [85] describes faces using Gabor filter responses in certain facial landmarks and a graph describing the spatial relations of these landmarks. The original Local Binary Pattern (LBP) operator was introduced by Ojala et al. as texture descriptor in an image [59]. In 2004, Ahonen, Hadid, and Pietikainen extended the LBP features for facial recognition [61]. The LBP operators use both shape and texture information to represent face images. In this

approach, a facial image is divided into small regions from which the LBP histograms are extracted and concatenated into a single feature histogram efficiently representing the face image [77]. The LBP operator has proven to be highly discriminative and is invariant to monotonic gray-level changes. Another advantage of the LBP operator is its computational efficiency making it suitable for demanding image analysis tasks.

7.2 Schematic Diagram of a Facial Identification System

A facial identification system has been developed as a part of this thesis to assess the performance of the novel feature descriptors. An identification system essentially recognizes a person by checking the entire template database for a match. It involves a "one to many" search. The system will make a match and subsequently identify the person or it will fail to make a match. Enrolment or training is the first stage of face recognition. The objective of the enrolment is to register the person into the system database. In the enrolment phase, the image of a person is captured by a sensor to produce a raw digital representation. The raw digital representation is then further processed by a feature extractor to generate a set of distinguishable features, called a template and stored in the system [78]. In the identification phase, features are extracted from the test image and compared with the features from the database. The result is either an enrolled user's identity or a warning message such as "person not identified". Schematic diagram of such a system has been shown in Figure 7.1.

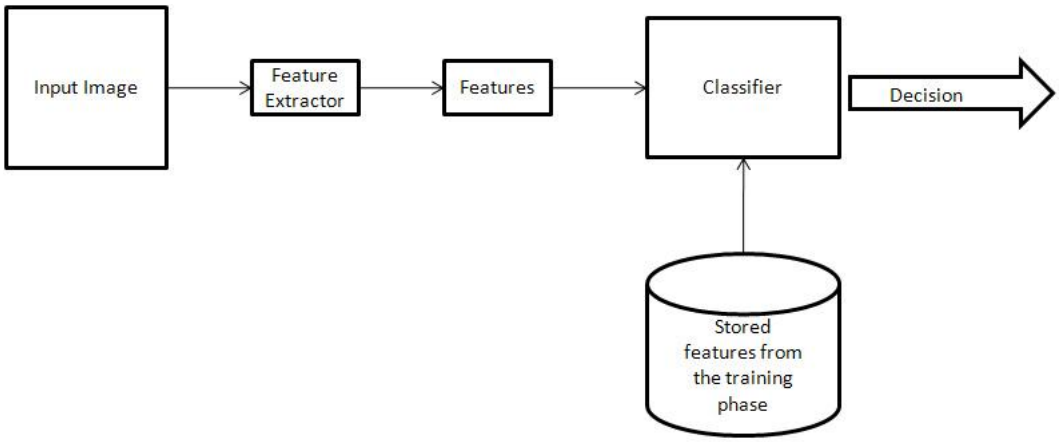


Figure 7.1: Schematic Diagram of a Facial Identification System

7.3 Local Binary Pattern (LBP) Feature Descriptors

The local binary pattern (LBP) operator, introduced by Ojala et al. [59], is a powerful local descriptor for describing image texture and has been used in many applications such as industrial visual inspection, image retrieval, automatic face recognition and detection. LBP texture descriptors can be used to build several local descriptors of the face and combine them to form a global feature. This approach was motivated by two reasons [59]: (1) the increasing popularity of local feature-based or hybrid approaches to face recognition [84] [86] due to their robustness against variations in pose or illumination than holistic methods and (2) tendency of holistic texture descriptors to average over the facial image area thereby loosing information about spatial relations. The basic LBP is a window based feature extractor where the texture descriptor is computed based on the neighborhood. The basic LBP operator has been discussed in section 4.5 of this thesis. It assigns a binary value to every neighboring pixel by thresholding it with respect to the central pixel. We assume that our input is an NxM 8-bit gray-level image, which can be represented as an NxM matrix I, each of whose elements satisfy $0 \leq I(x,y) \leq 2^8$. At a

given pixel position (x, y), the LBP operator is defined as an ordered set of binary comparisons (comparison implies calculation of the distance between the pixel intensities calculated by taking the arithmetic difference of pixel intensities) between the center pixel (x, y) and its surrounding pixels. Then the histogram of the labels can be used as a texture descriptor.

A histogram of the labeled image $f_l(x, y)$ can be defined as:

$$H_i = \sum_{x,y} I[f_l(x, y) = i], i = 0, \dots, n-1 \quad (7.1)$$

Where n is the number of different labels produced by the LBP operator and

$$I[A] = \begin{cases} 1, & A \; is \; true \\ 0, & A \; is \; false \end{cases} \quad (7.2)$$

This histogram contains information about the distribution of the local micro-patterns, such as edges, spots and flat areas, over the whole image and hence can be used to statistically describe image characteristics [87]. For more efficient face representation however, the facial image is divided into regions and feature descriptors are extracted from each region independently. Therefore if the image is divided into m regions, m local histograms are obtained. Let n be the size of each local LBP histogram, and then the combined LBP feature vector has the size $m \times n$. Thus with the spatially enhanced histogram, a description of the image on three different levels of locality is obtained. The LBP labels for the histogram contain information about the patterns on a pixel-level, the labels are summed over a small region to produce information on a regional level and the regional histograms are concatenated to build a global description of the face.

7.3.1 Uniform Local Binary Patterns

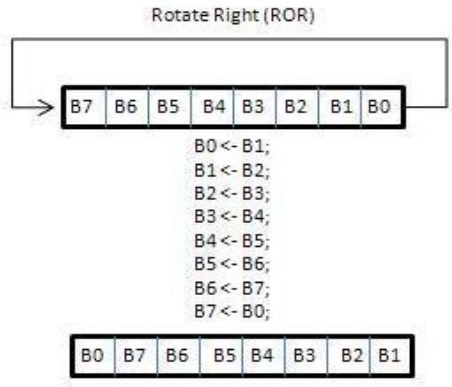
An extension to the original LBP operator is the uniform local binary patterns. A local binary pattern is called uniform if the binary pattern contains at most two bitwise transitions from 0 to 1 or vice versa when the bit pattern is considered circular. For example, the patterns 00000000 (0 transitions), 01110000 (2 transitions) and 11001111 (2 transitions) are uniform whereas the patterns 11001001 (4 transitions) and 01010011 (6 transitions) are not. In the computation of the LBP histogram, uniform patterns are used so that the histogram has a separate bin for every uniform pattern and all non-uniform patterns are assigned to a single bin. This is due to the fact that uniform patterns account for about 90 % of all patterns when using the (8, 1) neighborhood and for around 70 % in the (16, 2) neighborhood. Therefore, little information is lost by assigning all non-uniform patterns to a single arbitrary number. Since only 58 of the 256 possible 8 bit patterns are uniform, this enables significant space savings when building LBP histograms. To indicate the usage of two-transition uniform patterns, the superscript u2 is added to the LBP operator notation. Hence the LBP operator with a 2 pixel radius, 8 sampling points and uniform patterns is known as $LBP_{8,2}^{u2}$.

7.3.2 Rotation Invariant Local Binary Patterns

Another variant of the LBP operators designed originally for texture description is the rotation invariant local binary patterns [62]. The $LBP_{P,R}$ operator produces 2^P different output values. When the image is rotated, the gray values in the neighborhood will move along the perimeter of the circle surrounding the center pixel. Consequently the value of the LBP operators changes when the image is rotated. To remove the effect of rotation, this variant of LBPs were introduced in [62] as

$$LBP_{P,R}^{ri} = \min\left[ROR\left(LBP_{P,R}, i\right) \mid i = 0,1,\ldots,P-1\right] \tag{7.3}$$

Where $ROR\ (x, i)$ performs a circular bitwise right shift on a P bit number $x\ i$ times as shown in the schematic diagram below. So basically by applying bitwise right shifts, the goal is to find the pattern that has the maximum number of zeros in the most significant bits.



Rotation invariant patterns are often used in conjunction with the uniform patterns so that the length of the feature vector can be reduced.

7.4 Classifiers

An object detection and recognition system typically consists of 3 phases: training, testing and classification. The face recognition problem involves identifying a face from the database. The training facial images are preregistered in the system. Feature vectors are extracted from the training facial images and stored in the system. In the testing phase, feature vectors are extracted from the test image and compared with the feature vectors existing in the database. The test facial image is identified based on its similarity with images in the database. Usually, in face recognition, there are a number of face classes (representing individual person) and a few training images per class. Hence instead of using sophisticated classifier, a nearest-neighbor classification approach is used. The different types of dissimilarity measures that can be used are [83]:

Histogram Intersection:

$$D(S,M) = \sum_i \min(S_i, M_i) \tag{7.3}$$

Log-likelihood statistic:

$$L(S,M) = -\sum_i S_i log M_i \tag{7.4}$$

Chi square statistic χ^2

$$\chi^2(\mathrm{S},\mathrm{M}) = \sum_i \frac{(S_i - M_i)^2}{S_i + M_i} \tag{7.5}$$

Where S and M are the normalized histograms to be compared.

For most of our experiments, the chi square distance statistic has been used since it is an effective measure of similarity between a pair of histograms. Since the face is divided into smaller regions, these regions can be weighted differently depending on their contribution. Psychophysical findings indicate that some features like the eyes play more important role in human face recognition than others in terms of extra-personal variance [61]. The weighted chi square statistic is thus,

$$\chi^2(\mathrm{S},\mathrm{M}) = \sum_{i,j} w_j \frac{(S_{i,j} - M_{i,j})^2}{S_{i,j} + M_{i,j}} \tag{7.6}$$

Where the indices i and j refer to the $i-th$ bin corresponding to the $j-th$ local region and w_j is the weight for the region j. The weights can be determined by the methodology described in [84]. Here the training set is classified using one of the sub regions of the image at a time and the weights were assigned based on the rate of recognition. That is if the sub-region yielded a greater

rate of recognition, the weight associated with that region is selected to be higher than the others. The weights are however selected without utilizing an actual optimization procedure.

7.5 Logarithmic-LBP and PLIP-LBP Based Face Recognition

In this paper we have introduced two new types of LBP operators:

a) Logarithmic-LBP operators and
b) PLIP-LBP operators using the original image intensities and after applying logarithmic transform.

Further,

c) A third approach to LBP feature extraction has been introduced based on HVS image decomposition following the generic architecture introduced in this thesis. Figure 7.2 shows the LBP coded images after applying different types of LBP operators.

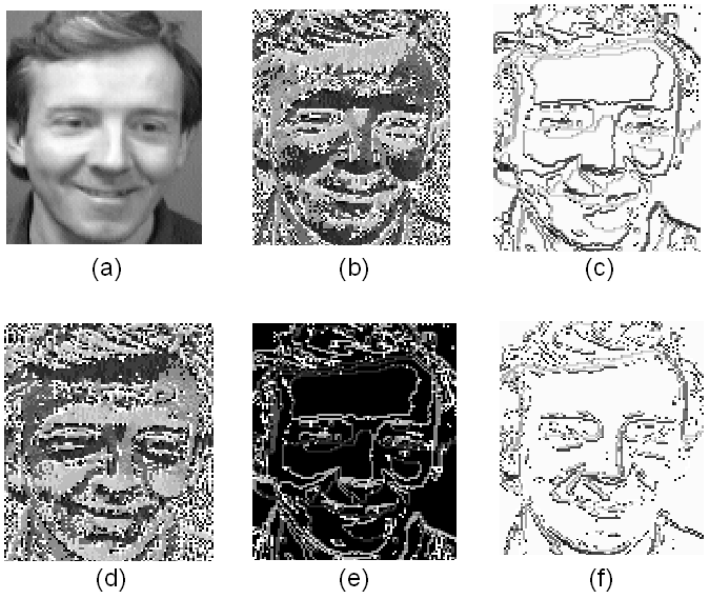


Figure 7.2: (a) Original Image and LBP coded images using (b) Classical LBP operator, (c) Logarithmic-LBP operator, (d) PLIP-LBP using original image intensities, (e) PLIP-LBP after logarithmic transform and (f) Logarithmic LBP on Weber image

From our experiments it has been found that the De-Vries Rose and the Saturation regions of HVS decomposition do not contain much useful information. Hence these regions are often fused with the Weber region. We refer to this fused region as the "Weber" image.

7.5.1 Logarithmic LBP Feature Descriptors

Logarithmic-LBP feature descriptors are introduced as a part of this thesis. In this approach instead of using the raw image pixels, a logarithmic transform is applied on the image and the then the LBP features are extracted in the logarithmic domain. Let $f(x, y)$ be the input image then the image in the logarithmic domain is given by:

$$I(x, y) = \log\left[eps + \left(\frac{f(x,y)}{\max(f(x,y))}\right)^{\alpha} . f(x, y)\right] \tag{7.7}$$

Where α is the parameter. In our experiments we have used several values of this parameter and obtained the results. The LBP feature extractor is then applied in the logarithmic domain to extract the LBP feature vectors. Uniform LBPs are used for face recognition as they outperform the other variants of LBP available. Here eps is a small number to avoid error in presence of zero image intensities. We have used feature fusion using this parameterized approach where the histograms generated from different values of the parameter alpha are concatenated to obtain an improvement in the average rate of recognition. Figure 7.3a show the classical LBP features extracted from the original image and the logarithmic-LBP features extracted after applying logarithmic transform to the image intensities. The value of α used here is 2.

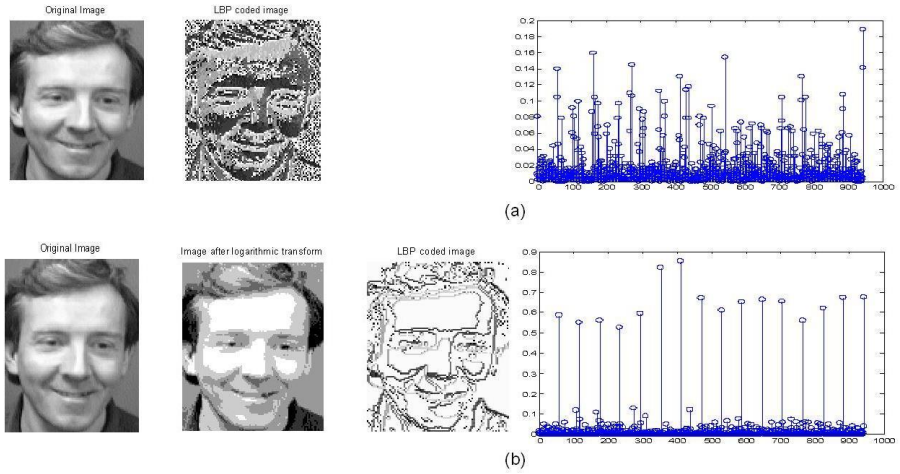


Figure 7.3a: (a) The original image, the LBP coded image and the LBP feature vector extracted from the original image; (b) Image after logarithmic transform, the logarithmic-LBP coded image and the extracted logarithmic-LBP feature vector

The flow diagram for face recognition based on logarithmic-LBP is shown in figure 7.3b.

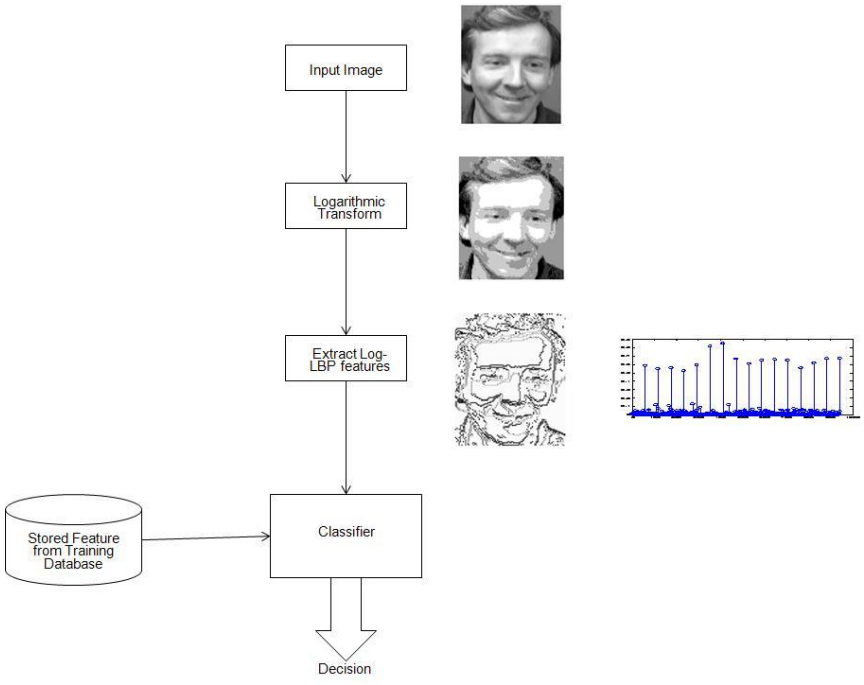


Figure 7.3b: Schematic Diagram: Log-LBP based face recognition

7.5.2 PLIP-LBP Feature Descriptors

In PLIP-LBP based feature descriptor, PLIP operations have been introduced while extracting the LBP feature extractors. The PLIP operators are consistent with the Weber's law and the saturation characteristics of the human visual system. The classical LBP operator assigns a label to every pixel of an image by thresholding a specified neighborhood of each pixel with the center pixel value and considering the result as a binary number. In PLIP-LBP, the calculation of the distance between the pixel intensities has been replaced by taking PLIP difference instead of the arithmetic difference. These operators are applied both on the original image intensities as well as after applying a logarithmic transform to the image intensities.

Hence the thresholding can now be defined in terms of the PLIP operators as:

$$I(x_i, y_i)\ \Theta\ I(x, y) = k(M)\frac{I(x_i,y_i)-I(x,y)}{k(M)-I(x,y)} \tag{7.8}$$

Where $I(x, y)$ is the center image pixel and $I(x_i, y_i)$ refers to the image pixels defined by the neighborhood. For example, for $\mathrm{LBP}_{8,2}^{\mathrm{u2}}$, i will range from 1 to 8. The value of the parameter $K(M)$ is chosen to be equal to 256. Figure 7.4a shows the PLIP-LBP coded image and the feature extracted by applying PLIP-LBP using the original image intensities and after applying logarithmic transform.

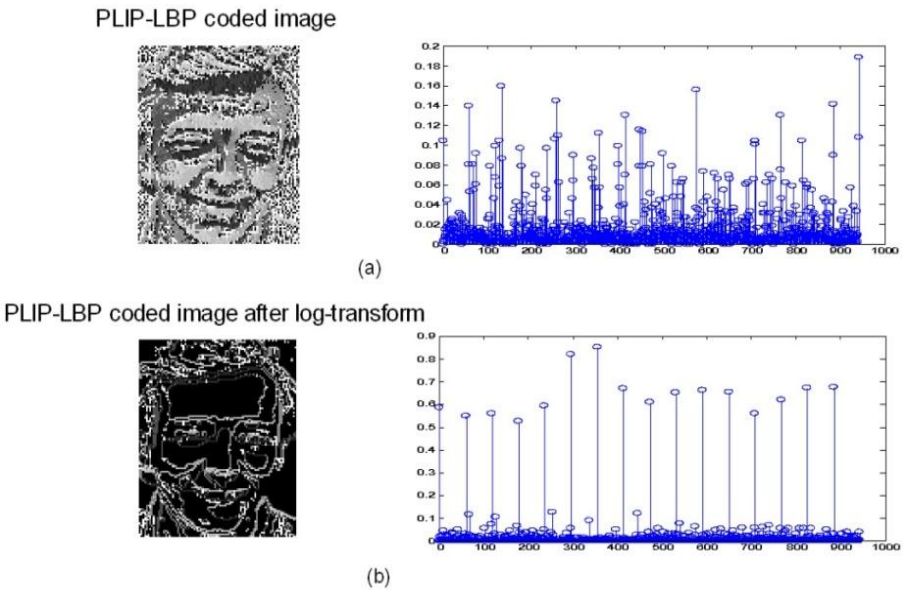


Figure 7.4a: (a) PLIP- LBP coded image and the PLIP-LBP feature vector extracted from the original image; (b) PLIP-LBP coded image after logarithmic transform and the extracted feature vector

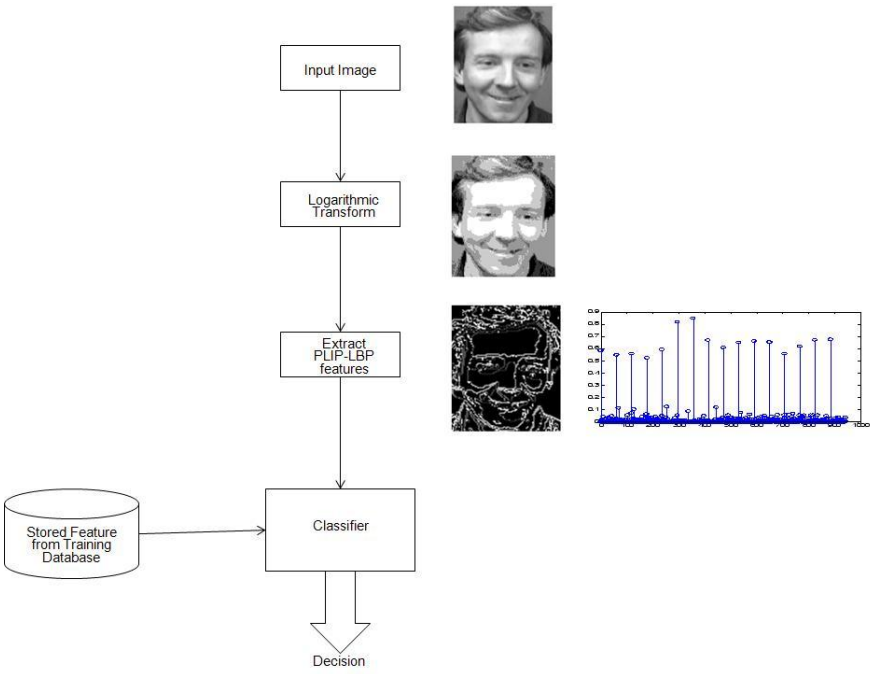


Figure 7.4b: Schematic Diagram: PLIP-LBP based face recognition

7.5.3 Logarithmic-LBP Feature Descriptors Based on HVS Image Decomposition

HVS based image decomposition is applied in this paper for extracting LBP histograms from the Weber image. LBP histograms obtained from the Weber image is concatenated with the LBP histogram obtained from the original image in the logarithmic domain following the generic architecture described in chapter 5. Thus the feature vector set can be expanded with the same set of training images thereby improving the rate of recognition. This method is particularly useful when the number of registered images per subject is less. Using HVS decomposition, we can expand the feature set using the same set of enrolled/training images. Using parameterization provides the added advantage of choosing different parameters for the original and the Weber images while extracting features. The parameters are chosen so as to give the best individual performances and hence a fusion of the features results in improved rates of recognition.

Figure 7.5a show the classical LBP features extracted from the Weber image and the logarithmic-LBP features extracted after applying logarithmic transform to the Weber image intensities. Since the De-Vries Rose and the Saturation regions of HVS do not contain significant information, we fuse these to images with the image from the Weber region and refer to the fused image as the Weber image. The value of α used here is 0.8.

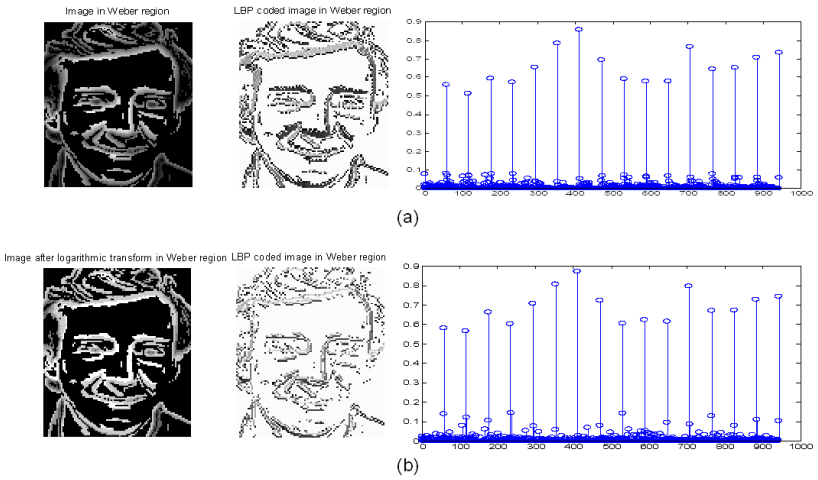


Figure 7.5a: (a) The Weber image, the LBP coded Weber image and the LBP feature vector extracted from the Weber image; (b) Weber Image after logarithmic transform, the LBP coded Weber image after logarithmic transform and the extracted LBP feature vector

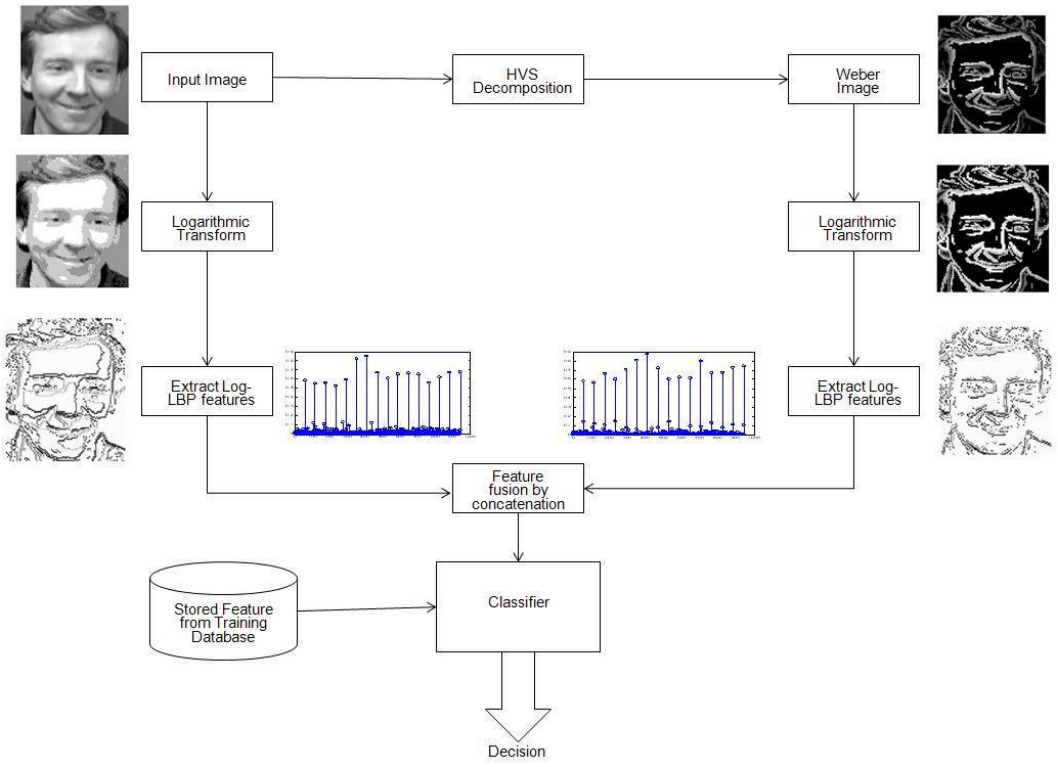


Figure 7.5b: Schematic Diagram: HVS-LBP based face recognition

Figure 7.6 shows the different types of modified LBP operators introduced in this thesis. The original image intensities are shown by the leftmost image in figure 7.6 (a) and (c). The images in (b) and (a) are derived after logarithmic transform of (a) after applying equation 7.7 and figure 7.6 (e) is the Weber image obtained from (a).

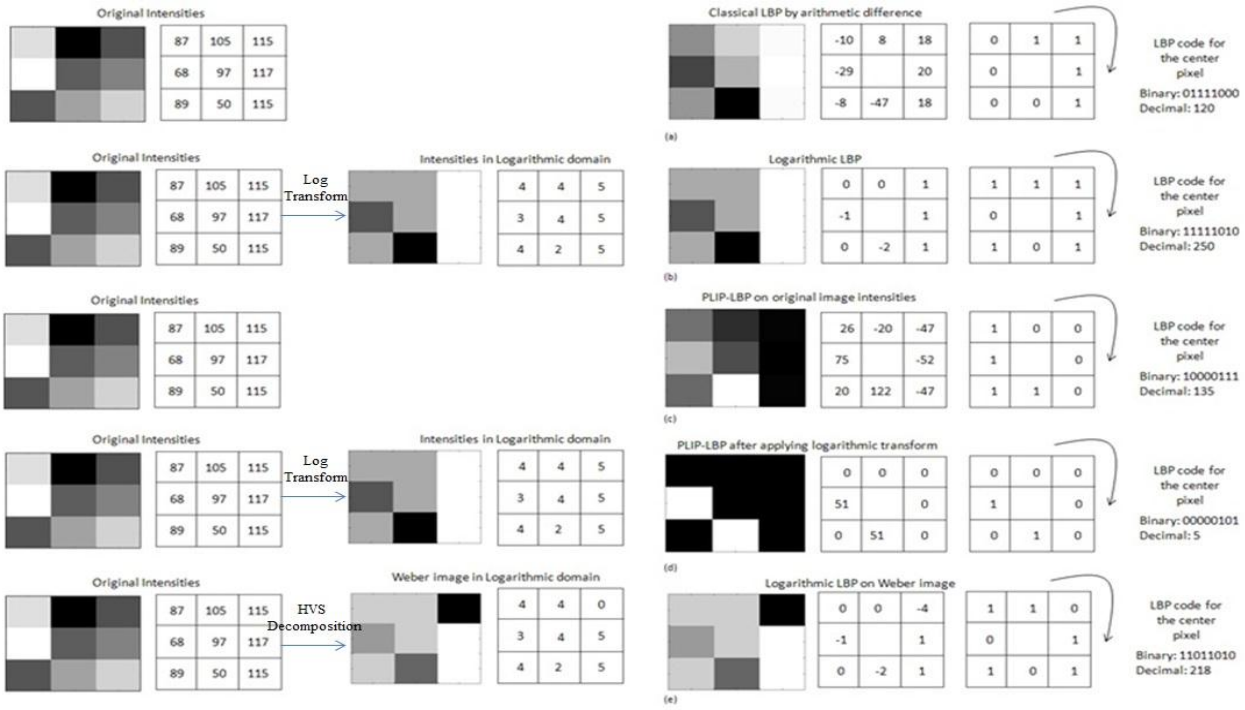


Figure 7.6: (a) The classical LBP operator (b) Logarithmic LBP operator (c) PLIP-LBP operator using the original image intensities (d) PLIP-LBP operators after applying logarithmic transform (e) Logarithmic LBP extracted from the Weber image

Thus the common steps of the algorithms used can be summarized as follows:

- Step 1: Input Image
- Step 2: Apply a Preprocessing algorithm for illumination normalization or noise reduction
- Step 3: Apply a modified LBP operator

- Step 4: Extract Local feature
- Step 5: Apply a classification procedure (Histogram intersection or the Chi square statistic)

7.6 Simulation and Results using AT&T Laboratories face database

To evaluate the performance of the proposed algorithm, AT&T Laboratories, Cambridge database were used. This database available in public domain and contains ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement).

- Two sets of experiments have been performed. In the first set, the first 5 images for each subject is used for training and registering and the remaining 5 images are used as test images.
- In the second set, 5 randomly chosen images are used for training and registration and the remaining 5 images are used for testing.
- Further, the number of images registered for a subject is varied from 1 to 5. Evidently the highest recognition rate is obtained when the number of registered images per subject is 5 and lowest when the number of registered images is 1.

7.6.1 Using Uniform LBP feature descriptors and Chi-square distance statistic

Uniform LBP feature descriptors are used in this section for feature extraction. Using uniform LBPs can reduce the size of the feature vector used and hence results is faster computations. Each facial image in the database has a size of 112X92. The image is divided into 16 equal

blocks of size 28X23. LBP features are extracted from each of these blocks and concatenated to form the final feature vector for the facial image. For a uint8 image 58 of the 256 possible 8 bit patterns are uniform and the non-uniform patterns are all put in the 59th bin. Hence the total length of the feature vector is 59*16 = 944. Chi-square distance statistic has been used for majority of our experiments. However we have also explored the use of other classifiers like histogram intersection or the log-likelihood statistic.

Registered training images per subject	Using classical LBP	Using Log-LBP with α=2	Using Log-LBP with α=1	Using Log-LBP with α=1.5	Using Log-LBP with α=0.9	Using Log-LBP with α=0.8
Number of images registered = 5	96.5	97	96.5	99	95	96
Number of images registered = 4	93	95.5	96	95	94.5	92
Number of images registered = 3	88	91.5	91.5	91.5	91.5	88.5
Number of images registered = 2	83.5	89.5	85.5	85	88.5	85.5
Number of images registered = 1	72	77.5	75	76	74	75

Table 7.1: Comparison of the rate of recognition of classical LBP vs. logarithmic LBP taking first 5 images per subject for training and remaining 5 images for testing. Using uniform LBP taking 8 points on a circular neighborhood of radius 1 and chi-square distance statistic

Table 7.1 shows the comparison of the rate of recognition using different values of the parameter α. From the above table it is found that when α = 2 there is an improved recognition rate under all testing conditions. Also it is evident that when all 5 images per subject are used for training, then α = 1.5 can attain a recognition rate of 99% out of the 200 testing images. We observe that a while with the value of 1.5 we can attain a recognition rate of 99% using all registered images, the average rate of recognition remains a bit higher with α = 2. Thus we introduce feature fusion after extracting logarithmic LBP using different values of alpha. Specifically we fuse the features for α = 1.5, 1.8 and 2, and compare the results with the classical LBP. The results are given in

Table 7.2. It is evident that the average rate has recognition has improved now for all testing conditions.

Registered training images per subject	Using classical LBP	Using fused logarithmic LBP with α=1.5, α=1.8 and α=2
Number of images registered = 5	96.5	98.5
Number of images registered = 4	93	96.5
Number of images registered = 3	88	93
Number of images registered = 2	83.5	90
Number of images registered = 1	72	79.5

Table 7.2: Comparison of the rate of recognition of classical LBP vs. fused logarithmic LBP taking first 5 images per subject for training and remaining 5 images for testing. Using uniform LBP taking 8 points on a circular neighborhood of radius 1 and chi-square distance statistic

We also performed tests by selecting randomly 5 images for training and the remaining 5 for testing. Results are given in table 7.3. Here value of $\alpha = 3$ gives the highest recognition rate under most of the testing conditions.

Registered training images per subject	Using classical LBP	Using Log-LBP with α=2	Using Log-LBP with α=1	Using Log-LBP with α=1.5	Using Log-LBP with α=0.9	Using Log-LBP with α=0.8	Using Log-LBP with α=3
Number of images registered = 5	95	94	94.5	95.5	96.5	95.5	97
Number of images registered = 4	94	94.5	93.5	94.5	96.5	96	95.5
Number of images registered = 3	91	90.5	91	90.5	93.5	92	93.5
Number of images registered = 2	83.5	85.5	86.5	84	88.5	86.6	89
Number of images registered = 1	73.5	81.5	80	78.5	79	77	79.5

Table 7.3: Comparison of the rate of recognition of classical LBP vs. logarithmic LBP taking randomly chosen 5 images per subject for training and remaining 5 images for testing. Using uniform LBP taking 8 points on a circular neighborhood of radius 1 and chi-square distance statistic

Table 7.4 compares the recognition rate using the Weber image of the HVS. In this case, an alpha value of 0.8/0.9 results in a better rate of recognition though there isn't a significant difference using the classical LBP and the log-LBP.

Registered training images per subject	Using classical LBP	Using Log-LBP with α=2	Using Log-LBP with α=1	Using Log-LBP with α=1.5	Using Log-LBP with α=0.9	Using Log-LBP with α=0.8	Using Log-LBP with α=0.5	Using Log-LBP with α=0.2
Number of images registered = 5	93.5	93.5	92	93	93.5	94	93.5	92
Number of images registered = 4	93	94	90.5	92.5	92.5	92.5	93	92.5
Number of images registered = 3	88.5	88.5	88	88.5	89	88	88	88
Number of images registered = 2	84.5	83.5	83.5	83.5	83	82	81	82
Number of images registered = 1	69.5	68.5	69	69.5	69.5	70	69.5	70

Table 7.4: Comparison of the rate of recognition of classical LBP vs. logarithmic LBP on Weber image taking first 5 images per subject for training and remaining 5 images for testing. Using uniform LBP taking 8 points on a circular neighborhood of radius 1 and chi-square distance statistic

Thus using log-LBP feature extraction on the Weber image alone does not result in the improvement of the recognition rate. Hence the log-LBP features extracted from the Weber image are fused with the log-LBP features extracted from the original image. One of the advantages of using a parameterized logarithmic transform is that we can choose different values of α while extracting features from the original and the Weber image. The values are chosen so as to give the maximum possible rate of recognition when applied independently on the original and the Weber image. From table 7.5 it is evident that fused log-LBP using HVS results in a better rate of recognition under all training conditions. The parameter α is selected as 2 when doing a logarithmic transform of the original image and a value equal to 0.8 is selected when extracting the log-LBP feature vectors from the Weber image for the initial testing configurations i.e. using the first 5 images for each subject for training and using the remaining 5 images for testing. For the second testing configuration where randomly selected 5 images were used for training and the remaining 5 images were used for testing, the parameter α is selected as 2 when doing a logarithmic transform of the original image and a value equal to 0.8 is selected

when extracting the log-LBP feature vectors from the Weber image because these values individually result into a high recognition rates when compared to other values of the parameter.

Registered training images per subject	Testing configuration #1 Using classical LBP	Testing configuration #1 Using fused Log-LBP with α=2 for the original image and α=0.8 for the Weber image	Testing configuration #2 Using classical LBP	Testing configuration #2 Using fused Log-LBP with α=3 for the original image and α=0.8 for the Weber image
Number of images registered = 5	96.5	98.5	95	97.5
Number of images registered = 4	93	96.5	94	95
Number of images registered = 3	88	93	91	92.5
Number of images registered = 2	83.5	88.5	83.5	88
Number of images registered = 1	72	78	73.5	79.5

Table 7.5: Comparison of the rate of recognition of classical LBP vs. fused log-LBP (log-LBP feature vector from the original image is fused with the log-LBP feature vector obtained from the Weber image).Test configuration #1 takes first 5 images per subject for training and last 5 images for testing. Test configuration #2 takes randomly selected 5 images per subject for training and remaining 5 images for testing. Using uniform LBP taking 8 points on a circular neighborhood of radius 1 and chi-square distance statistic

Registered training images per subject	Using classical LBP	Using PLIP-LBP	Using PLIP-LBP after applying logarithmic transform with α=2
Number of images registered = 5	96.5	96.5	97.5
Number of images registered = 4	93	93	95.5
Number of images registered = 3	88	88.5	92
Number of images registered = 2	83.5	84	90
Number of images registered = 1	72	71.5	79.5

Table 7.6: Comparison of the rate of recognition of classical LBP vs. PLIP-LBP taking first 5 images per subject for training and remaining 5 images for testing. Using uniform LBP taking 8 points on a circular neighborhood of radius 1 and chi-square distance statistic

Registered training images per subject	Using classical LBP	Using PLIP-LBP	Using PLIP-LBP after applying logarithmic transform with α=3
Number of images registered = 5	95	95	97
Number of images registered = 4	94	94	95.5
Number of images registered = 3	91	91	94
Number of images registered = 2	83.5	83.5	89
Number of images registered = 1	73.5	73	79.5

Table 7.7: Comparison of the rate of recognition of classical LBP vs. PLIP-LBP taking randomly selected 5 images per subject for training and remaining 5 images for testing. Using uniform LBP taking 8 points on a circular neighborhood of radius 1 and chi-square distance statistic

We have also used PLIP operators for computing the distance between the intensities of the image pixels in LBP. Tables 7.6 and 7.7 compare the recognition rates using PLIP-LBP.

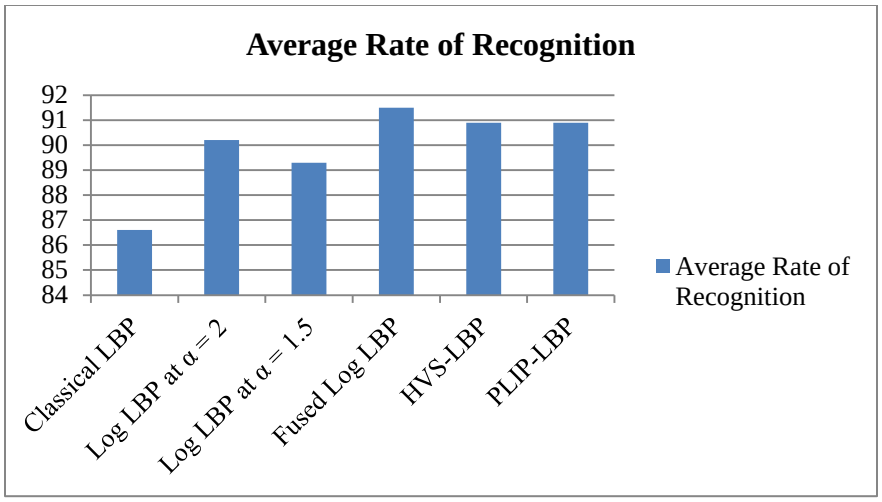


Figure 7.7: Comparison of average recognition rate taking first 5 images per subject for training and remaining 5 images for testing. Using uniform LBP with Chi-square statistic

Figure 7.7 compares the average rate of recognition using different LBP operators. The average rate of recognition is calculated by taking the average of the recognition rates under different testing conditions that is by varying the number of registered training images from 1 to 5. Similarly figure 7.8 compares the average rate when taking randomly selected 5 images for making the training database and remaining 5 images for testing.

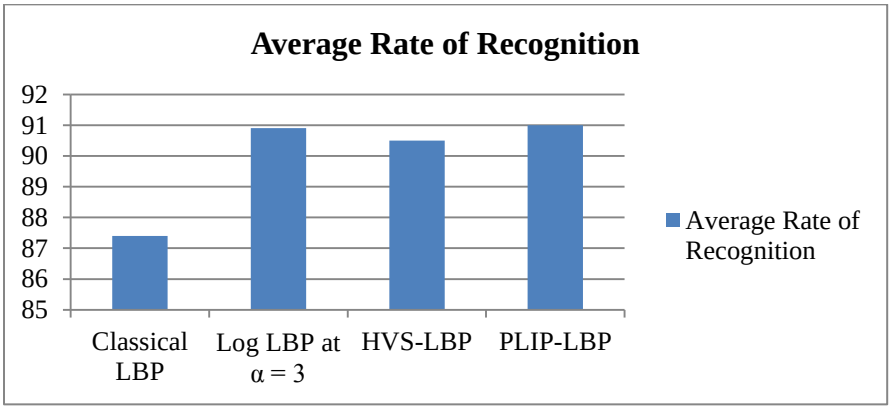


Figure 7.8: Comparison of average recognition rate taking randomly selected 5 images per subject for training and remaining 5 images for testing. Using uniform LBP with Chi-square statistic

7.6.2 Using Uniform LBP feature descriptors and Histogram Intersection statistic

Although chi-square distance statistic has been used for most of our experiments, we have also explored the results using other classifiers like the histogram intersection and the log-likelihood statistic. While comparable results have been obtained using the histogram intersection statistic, using the log-likelihood statistic for recognition however the recognition rates have been found to be lower (83.5 % compared to 96.5% using the chi-square distance statistic). Table 7.8 compares the rates of recognition using the histogram intersection statistic.

Registered training images per subject	Using classical LBP	Using Log-LBP with α=2	Using Log-LBP with α=2.5	Using PLIP-LBP	Using PLIP-LBP after applying logarithmic transform with α=2	Using PLIP-LBP after applying logarithmic transform with α=2.5
Number of images registered = 5	95	96.5	97	94.5	96	97
Number of images registered = 4	90	92.5	93	90	92.5	93
Number of images registered = 3	86	86.5	88	86	87	88.5
Number of images registered = 2	82	81.5	84.5	82	82.5	84
Number of images registered = 1	71.5	71.5	72.5	71.5	71.5	72.5

Table 7.8: Comparison of the rates of recognition using the histogram intersection distance statistic using different types of LBP operators. First 5 images are used for training and the remaining 5 images for testing. Using uniform LBP taking 8 points on a circular neighborhood of radius 1

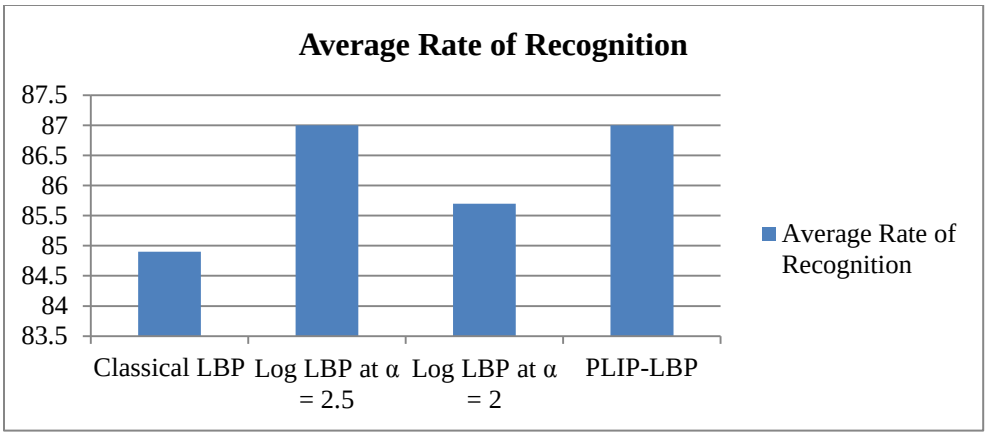


Figure 7.9: Comparison of average recognition rate taking first 5 images per subject for training and remaining 5 images for testing. Using uniform LBP with histogram intersection statistic

7.6.3 Using Uniform LBP feature descriptors and weighted Chi-square statistic

The weighted chi-square statistic associates different weights to the LBP histograms obtained from the sub-blocks in the image. We placed more weightage to the eyes and mouth than the remaining portions of the face while doing the classification. We present the results in this section. We show that out modified LBP operators outperforms the performance of the classical LBP operators using different types of classifiers. The results also show the consistency in performance of the modified LBP operators developed as a part of this thesis.

Registered training images per subject	Using classical LBP	Using Log-LBP with α=2	Using Log-LBP with α=1.5	Using Log-LBP with α=1.8	Using Log-LBP with α=2.2
Number of images registered = 5	95	97	95	97.5	95
Number of images registered = 4	91.5	96	91.5	95	94.5
Number of images registered = 3	85.5	91	87.5	90.5	90.5
Number of images registered = 2	80	87.5	82.5	85	87
Number of images registered = 1	65.5	74	72.5	74.5	75

Table 7.9: Comparison of the rate of recognition of classical LBP vs. logarithmic LBP taking first 5 images per subject for training and remaining 5 images for testing. Using uniform LBP taking 8 points on a circular neighborhood of radius 1 and weighted chi-square distance statistic

We also combine the feature extracted using different values of the parameter alpha and fuse the feature vectors.

Registered training images per subject	Using classical LBP	Using fused logarithmic LBP with α=1.5, α=1.8 and α=2	Using fused logarithmic LBP with α=2.2, α=1.8 and α=2
Number of images registered = 5	95	98	97.5
Number of images registered = 4	91.5	95.5	95
Number of images registered = 3	85.5	92	91
Number of images registered = 2	80	87	87
Number of images registered = 1	65.5	74	77

Table 7.10: Comparison of the rate of recognition of classical LBP vs. fused logarithmic LBP taking first 5 images per subject for training and remaining 5 images for testing. Using uniform LBP taking 8 points on a circular neighborhood of radius 1 and chi-square distance statistic

We have also applied HVS based decomposition and fused the feature extracted from the original and the Weber images.

Registered training images per subject	Using classical LBP	Using fused Log-LBP with α=2 for the original image and α=0.8 for the Weber image
Number of images registered = 5	95	97.5
Number of images registered = 4	91.5	97
Number of images registered = 3	85.5	91.5
Number of images registered = 2	80	86.5
Number of images registered = 1	65.5	73.5

Table 7.11: Comparison of the rate of recognition of classical LBP vs. fused log-LBP (log-LBP feature vector from the original image is fused with the log-LBP feature vector obtained from the Weber image) taking first 5 images per subject for training and last 5 images for testing. Using uniform LBP taking 8 points on a circular neighborhood of radius 1 and weighted chi-square distance statistic

Finally a comparison of the classical LBP with PLIP-LBP operators is presented.

Registered training images per subject	Using classical LBP	Using PLIP-LBP	Using PLIP-LBP after applying logarithmic transform with α=2
Number of images registered = 5	95	95	97.5
Number of images registered = 4	91.5	91.5	95.5
Number of images registered = 3	85.5	85.5	90.5
Number of images registered = 2	80	80.5	86.5
Number of images registered = 1	65.5	66	74

Table 7.12: Comparison of the rate of recognition of classical LBP vs. PLIP-LBP taking first 5 images per subject for training and remaining 5 images for testing. Using uniform LBP taking 8 points on a circular neighborhood of radius 1 and weighted chi-square distance statistic

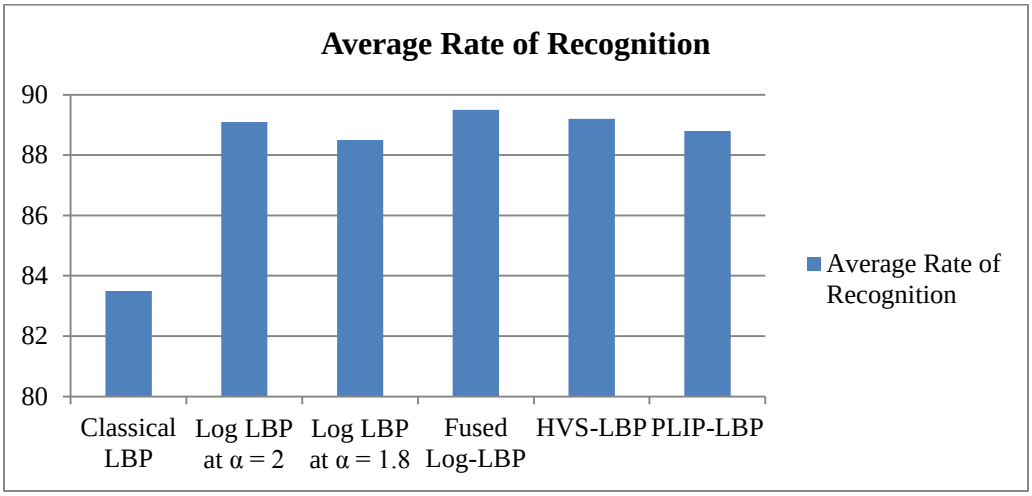


Figure 7.10: Comparison of average recognition rate taking first 5 images per subject for training and remaining 5 images for testing. Using uniform LBP with weighted Chi-square statistic

7.6.4 Using Rotation Invariant LBP features with Weighted Chi-square statistic

We have also performed experiments with the rotation invariant LBP feature descriptors. Using the classical rotation invariant LBPs for feature extraction results in the decrease or recognition rates compared to the uniform LBPs. However using our modified LBP operators, the recognition rates, though slightly less as compared to the modified uniform LBPs, are comparable. This proves the consistency of our proposed feature vectors. A summary of results have been presented in this chapter.

Registered training images per subject	Using classical LBP	Using Log-LBP with α=2	Using Log-LBP with α=1.8	Using Log-LBP with α=2.2	Using Log-LBP with α=2.5	Using Log-LBP with α=1.9
Number of images registered = 5	86	93	95.5	93.5	95	94.5
Number of images registered = 4	83	90	92.5	92	90.5	92
Number of images registered = 3	77	87	88	88	86	88
Number of images registered = 2	73	84	84	84.5	82.5	83.5
Number of images registered = 1	62	75	72.5	72	72	69.5

Table 7.13: Comparison of the rate of recognition of classical LBP vs. logarithmic LBP taking first 5 images per subject for training and remaining 5 images for testing. Using rotation invariant LBP taking 8 points on a circular neighborhood of radius 2 and weighted chi-square distance statistic

A feature fusion taking the best values of alpha yields the following results.

Registered training images per subject	Using classical LBP	Using fused logarithmic LBP with α=2.2, α=1.8 and α=2
Number of images registered = 5	86	96
Number of images registered = 4	83	93.5
Number of images registered = 3	77	91.5
Number of images registered = 2	73	87
Number of images registered = 1	62	77

Table 7.14: Comparison of the rate of recognition of classical LBP vs. fused logarithmic LBP taking first 5 images per subject for training and remaining 5 images for testing. Using rotation invariant LBP taking 8 points on a circular neighborhood of radius 2 and weighted chi-square distance statistic

7.6.5 Using Rotation Invariant LBP features with non-weighted Chi-square statistic

Rotation invariant LBPs are also used with the non-weighted Chi-square distance statistic. In this section we present some of the results comparing the performances of the classical and the modified LBP operators.

Registered training images per subject	Using classical LBP	Using PLIP-LBP	Using PLIP-LBP after applying logarithmic transform with α=1.8
Number of images registered = 5	87.5	87.5	95.5
Number of images registered = 4	83.5	83.5	93.5
Number of images registered = 3	78.5	78.5	89
Number of images registered = 2	76.5	77	86
Number of images registered = 1	64	64.5	76

Table 7.15: Comparison of the rate of recognition of classical LBP vs. PLIP-LBP taking first 5 images per subject for training and remaining 5 images for testing. Using rotation invariant LBP taking 8 points on a circular neighborhood of radius 2 and chi-square distance statistic

We present the results using HVS decomposition and feature extraction in Table 7.16

Registered training images per subject	Using classical LBP	Using fused Log-LBP with α=1.8 for the original image and α=0.8 for the Weber image
Number of images registered = 5	87.5	94.5
Number of images registered = 4	83.5	94
Number of images registered = 3	78.5	89
Number of images registered = 2	76.5	86.5
Number of images registered = 1	64	73.5

Table 7.16: Comparison of the rate of recognition of classical LBP vs. fused log-LBP (log-LBP feature vector from the original image is fused with the log-LBP feature vector obtained from the Weber image) taking first 5 images per subject for training and last 5 images for testing. Using rotation invariant LBP taking 8 points on a circular neighborhood of radius 2 and chi-square distance statistic

7.6.6 Using Rotation Invariant Uniform LBP with Chi-square distance statistic

The rotation invariant LBPs can be uniform. In this case all the non-uniform patterns are placed in a single bin and the uniform patterns are placed in their respective bins. We have used the Chi-square distance statistic to obtain the relative performance of our modified LBP operators with respect to the classical LBP operators.

Registered training images per subject	Using classical LBP	Using Log-LBP with α=2	Using Log-LBP with α=1.8	Using Log-LBP with α=2.2
Number of images registered = 5	83.5	94	94	94
Number of images registered = 4	80	91.5	91	92
Number of images registered = 3	76.5	88.5	86.5	89.5
Number of images registered = 2	71.5	84	82.5	86.5
Number of images registered = 1	60	73	73	73.5

Table 7.17: Comparison of the rate of recognition of classical LBP vs. logarithmic LBP taking first 5 images per subject for training and remaining 5 images for testing. Using rotation invariant uniform LBP taking 8 points on a circular neighborhood of radius 2 and chi-square distance statistic

Registered training images per subject	Using classical LBP	Using fused logarithmic LBP with α=2.2, α=1.8 and α=2
Number of images registered = 5	83.5	96
Number of images registered = 4	80	94
Number of images registered = 3	76.5	90
Number of images registered = 2	71.5	86
Number of images registered = 1	60	75

Table 7.18: Comparison of the rate of recognition of classical LBP vs. fused logarithmic LBP taking first 5 images per subject for training and remaining 5 images for testing. Using rotation invariant uniform LBP taking 8 points on a circular neighborhood of radius 2 and chi-square distance statistic

Registered training images per subject	Using classical LBP	Using PLIP-LBP	Using PLIP-LBP after applying logarithmic transform with α=2.2
Number of images registered = 5	83.5	83.5	94
Number of images registered = 4	80	79.5	92
Number of images registered = 3	76.5	76.5	89.5
Number of images registered = 2	71.5	71.5	86
Number of images registered = 1	60	60	72.5

Table 7.19: Comparison of the rate of recognition of classical LBP vs. PLIP-LBP taking first 5 images per subject for training and remaining 5 images for testing. Using rotation invariant uniform LBP taking 8 points on a circular neighborhood of radius 2 and chi-square distance statistic

Registered training images per subject	Using classical LBP	Using fused Log-LBP with α=2.2 for the original image and α=0.8 for the Weber image	Using fused Log-LBP with α=2.2 for the original image and α=1 for the Weber image
Number of images registered = 5	83.5	95	95.5
Number of images registered = 4	80	91.5	92.5
Number of images registered = 3	76.5	88.5	89
Number of images registered = 2	71.5	83	82.5
Number of images registered = 1	60	71.5	71.5

Table 7.20: Comparison of the rate of recognition of classical LBP vs. fused log-LBP (log-LBP feature vector from the original image is fused with the log-LBP feature vector obtained from the Weber image) taking first 5 images per subject for training and last 5 images for testing. Using rotation invariant uniform LBP taking 8 points on a circular neighborhood of radius 2 and chi-square distance statistic

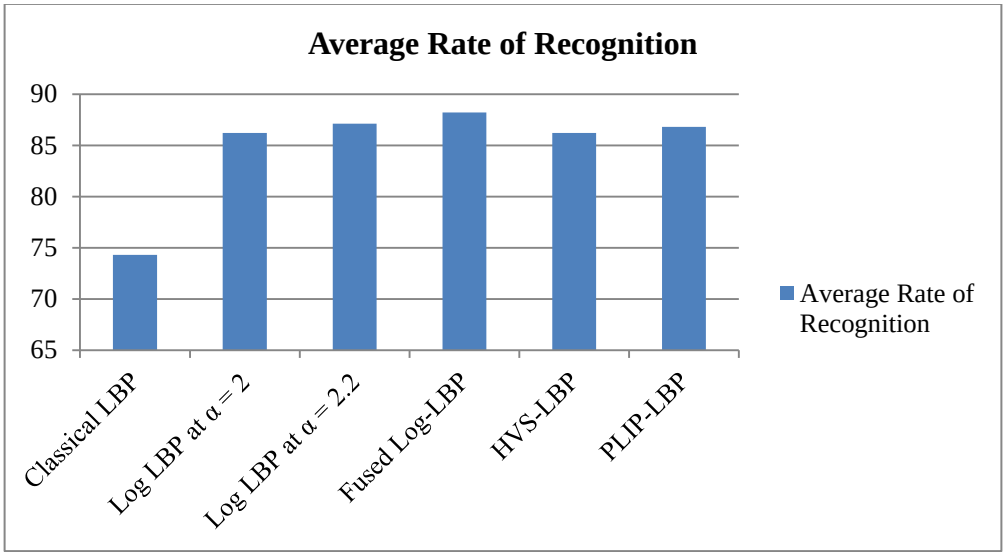


Figure 7.11: Comparison of average recognition rate taking first 5 images per subject for training and remaining 5 images for testing. Using rotation invariant uniform LBP with Chi-square statistic

7.7 Simulation and Results using the Yale face database

We have also performed experiments using the Yale face database. This database is also available publicly and consists of 15 subjects each having 11 images. For simplicity we have taken first 5 images per subject for training and the next 5 images for testing. We have used Chi-square distance statistic for performing classification.

Registered training images per subject	Using classical LBP	Using Log-LBP with α=1	Using Log-LBP with α=1.2	Using Log-LBP with α=0.9
Number of images registered = 5	96	98.67	96	98.67
Number of images registered = 4	96	97.33	96	97.33
Number of images registered = 3	96	97.33	96	97.33
Number of images registered = 2	96	97.33	94.67	97.33
Number of images registered = 1	89.33	87	85.33	85.33

Table 7.21: Comparison of the rate of recognition of classical LBP vs. logarithmic LBP taking first 5 images per subject for training and next 5 images for testing. Using uniform LBP taking 8 points on a circular neighborhood of radius 1 and chi-square distance statistic

Thus we see that the performance of our feature vectors is consistent across different databases and using different classification methods. We have also used HVS decomposition to obtain an enhanced feature vector set. The results obtained are presented in table 7.22.

Registered training images per subject	Using classical LBP	Using fused Log-LBP with α=1 for the original image and using features from the Weber image directly
Number of images registered = 5	96	98.67
Number of images registered = 4	96	100
Number of images registered = 3	96	100
Number of images registered = 2	96	100
Number of images registered = 1	89.33	85.34

Table 7.22: Comparison of the rate of recognition of classical LBP vs. fused log-LBP (log-LBP feature vector from the original image is fused with the LBP feature vector obtained from the Weber image) taking first 5 images per subject for training and next 5 images for testing. Using uniform LBP taking 8 points on a circular neighborhood of radius 1 and chi-square distance statistic

We also present the results using the PLIP-LBP operators.

Registered training images per subject	Using classical LBP	Using PLIP-LBP	Using PLIP-LBP after applying logarithmic transform with α=1
Number of images registered = 5	96	96	98.67
Number of images registered = 4	96	96	97.33
Number of images registered = 3	96	96	97.33
Number of images registered = 2	96	96	96
Number of images registered = 1	89.33	89.33	87

Table 7.23: Comparison of the rate of recognition of classical LBP vs. PLIP-LBP taking first 5 images per subject for training and next 5 images for testing. Using uniform LBP taking 8 points on a circular neighborhood of radius 1 and chi-square distance

Figure 7.12 summarizes the results.

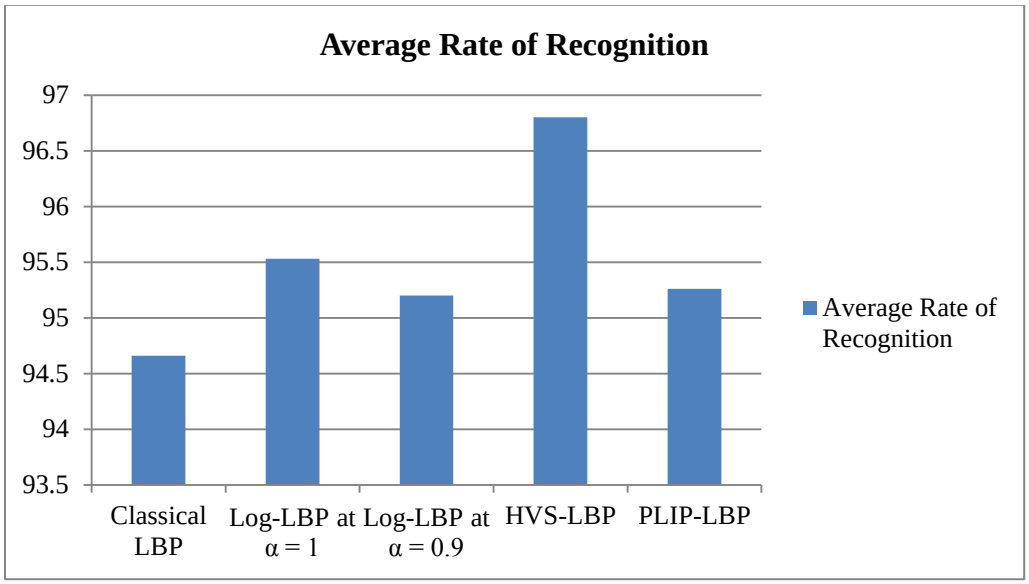


Figure 7.12: Comparison of average recognition rate taking first 5 images per subject for training and nest 5 images for testing. Using uniform LBP with Chi-square statistic

7.8 Examples of Recognition

In this section we present some examples of face recognition using the AT&T and the Yale face database using the modified LBP operators. We compare with face recognition using the classical LBP operators.

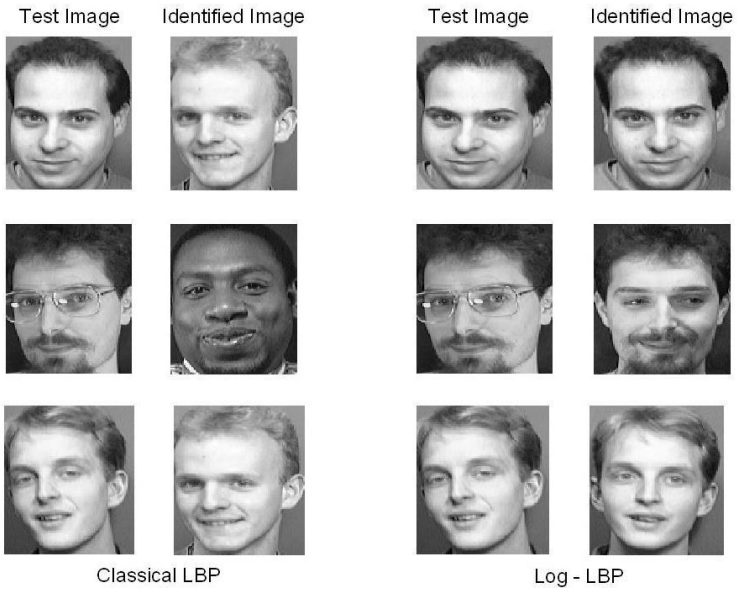


Figure 7.13: Comparison of face recognition using classical and logarithmic LBP using 5 images for training from AT&T face database

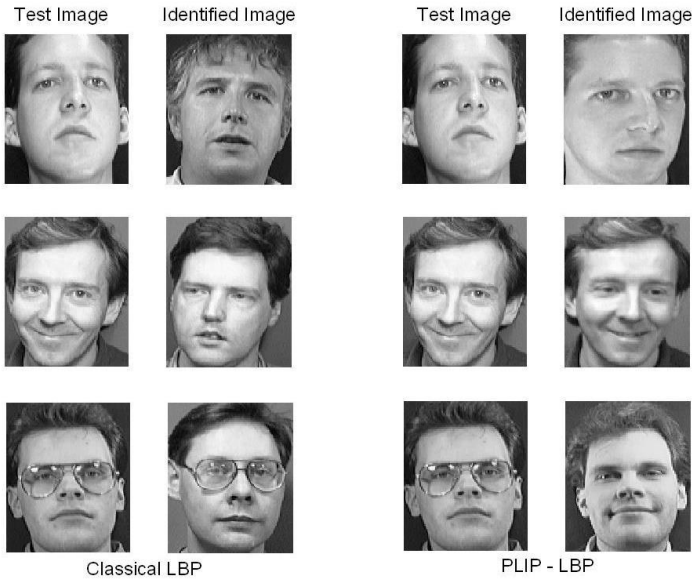


Figure 7.14: Comparison of face recognition using classical and PLIP-LBP using 3 images for training from AT&T face database

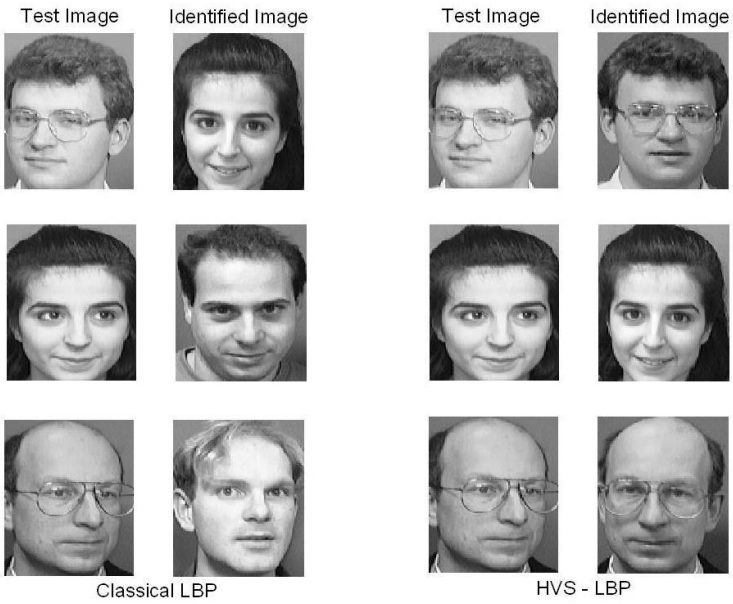


Figure 7.15: Comparison of face recognition using classical and HVS-LBP using 1 image for training from AT&T face database

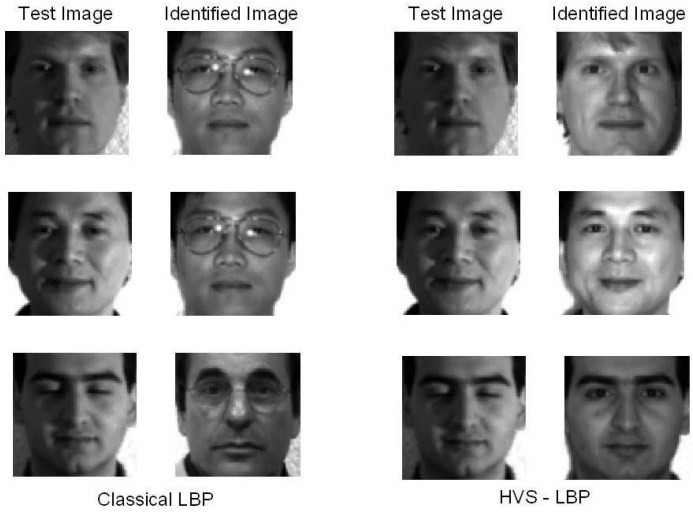


Figure 7.16: Comparison of face recognition using classical and HVS-LBP using 4 images for training from Yale face database

Figures 7.13 to 7.16 show the missed recognitions using classical LBP and how using the modified LBP operators, true recognitions can be obtained.

7.9 Analysis and Discussion

In this section we'll discuss the results presented in section 7.6 and 7.7. In section 7.6 we have used different variants of LBP in conjunction with different types of classifiers. From the results it can be seen that using the modified LBP operators, improved rates of recognition have been obtained as compared to the classical LBP operators using different LBP variants and different classifiers.

Table 7.25 compares the performances of the modified LBP operators. The classifier is constant here and the chi-square distance statistic is used here.

Registered training images per subject	Using uniform fused logarithmic LBP with α=1.5, α=1.8 and α=2	Using uniform HVS-LBP with α=2 for the original image and α=0.8 for the Weber image	Using uniform PLIP-LBP after applying logarithmic transform with α=2	Using rotation invariant uniform Log-LBP with α=2	Using rotation invariant PLIP-LBP after applying logarithmic transform with α=1.8
Number of images registered = 5	98.5	98.5	97.5	94	95.5
Number of images registered = 4	96.5	96.5	95.5	91.5	93.5
Number of images registered = 3	93	93	92	88.5	89
Number of images registered = 2	90	88.5	90	84	86
Number of images registered = 1	79.5	78	79.5	73	76

Table 7.24: Comparison of the rate of recognition for the modified LBP operators using Chi-square distance statistic

Thus we see that using the uniform LBP results is better rates of recognition when compared to the rotation invariant or the rotation invariant uniform LBP. Comparing columns 1, 2 and 3 we see that fused Log-LBP and HVS-LBP shows comparable performance. Using uniform LBP also helps to reduce the dimensionality of the feature vector and hence results in faster computations. Rotation invariant LBPs have found to perform well for texture classification, however uniform LBPs have performed well for face recognition. However we also show that the modified LBP operators are more or less consistent across different LBP variants.

We'll now compare the performances using different types of nearest neighbor classification approaches. The log-likelihood statistic usually fails to perform as good as the other classification approaches for LBP based face recognition (83.5 % compared to 96.5% using the chi-square distance statistic). In table 7.25 we compare the performances using the different nearest neighbor classifiers.

Registered training images per subject	Using uniform fused logarithmic LBP with α=1.5, α=1.8 and α=2 and chi-square distance statistic	Using uniform HVS-LBP with α=2 for the original image and α=0.8 for the Weber image and chi-square distance statistic	Using uniform Log-LBP with α=2.5 and histogram intersection distance statistic	Using uniform PLIP-LBP after applying logarithmic transform with α=2.5 and histogram intersection distance statistic	Using uniform HVS-LBP with α=2 for the original image and α=0.8 for the Weber image and weighted chi-square distance statistic	Using uniform Log-LBP with α=2 and weighted chi-square distance statistic
Number of images registered = 5	98.5	98.5	97	97	97.5	97
Number of images registered = 4	96.5	96.5	93	93	97	96
Number of images registered = 3	93	93	88	88.5	91.5	91
Number of images registered = 2	90	88.5	84.5	84	86.5	87.5
Number of images registered = 1	79.5	78	72.5	72.5	73.5	74

Table 7.25: Comparison of the rate of recognition for the modified LBP operators across different classifiers

Thus from the results presented in table 7.25 we see that the non-weighted chi-square distance usually results in a better average rate of recognition when we are taking into account all testing configurations. The weights in this case were chosen so as to give more weight to the region near the eyes or mouth. A more structural approach can be used while determining the weights. The training set can be classified using one of the sub regions of the image at a time and the weights can be assigned based on the rate of recognition. That is if the sub-region yielded a greater rate of recognition, the weight associated with that region is selected to be higher than the others. Thus we can conclude that using uniform log-LBP or the uniform HVS-LBP along with the chi-square distance statistic gives the best rate of recognition among all other approaches.

7.10 Execution Time and Memory Usage

The execution time for different LBP operators are presented in table 7.26. For computing the total time of execution, 200 test images were run through the system using uniform LBP feature vectors.

Processing Time Classical LBP	Processing Time fused Log-LBP	Processing Time PLIP-LBP	Processing Time HVS-LBP
32.7 seconds	53.2 seconds	51.7	51.4

Table 7.26: Comparison of the execution time for different types of LBP operators

Thus we see that using the modified LBP operators, the time of execution is about 1.5 times that taken by the classical LBP operator. There is a computational overhead involved while doing logarithmic transform. Also in the fused LBP approaches, features are extracted from multiple images at the same time. For example when using the HVS based approach, the LBP features are extracted using the original image and the Weber image thereby taking twice the memory than the classical approach in the feature extraction stage. Thus if there are "N" ALU operations involved in the feature extraction step, then HVS-LBP would require "2N" ALU operations and fused log-LBP using three values of the parameter alpha would require "3N" ALU operations. However since extraction of feature vectors from these different images are independent of one another, parallel processing can be leveraged using distributed systems.

7.11 Summary

In this section we summarize the ideas presented in this chapter:

- Three types of novel feature extraction methods have been introduced in this chapter.
- Using all three modified LBP operators we have shown how better recognition rates can be obtained as compared to the classical LBP feature extraction algorithm.

- Our modified LBP operators are based on the Weber's law and the saturation characteristics of the human visual system
- We have tested using two different configuration of the training set and obtained comparable performances.
- We have also varied the number of enrolled images per subject and obtained improved average rates of recognition.
- Thus by incorporating the human visual response and the logarithmic image processing framework which emulates the human visual characteristics we have obtained improved feature vectors that can be applied towards a variety of practical applications like face recognition.

CONCLUSION

A new framework for object detection and recognition has been introduced in this thesis. It has been shown how HVS based image processing can be successfully applied towards image processing applications like edge detection, feature extraction, object detection and recognition. Feature fusion has been incorporated in the framework. It can be concluded that by decomposing image based on the human visual sensitivity, more information can be extracted out of the image. Two systems have been developed which take into account HVS based image decomposition and thereby have a better rate of detection and recognition compared to the non-HVS based systems. In case of eye detection, decomposing an image also helps when the regions near the eyes have shadows or when one of the eyes is darker than the other. This is because HVS based decomposition takes into account the background intensity of the individual image pixels when choosing the thresholds for different HVS regions. Thus darker and lighter regions of the image have different thresholds which tend to bring out the pertinent details in a more effective manner.

Three novel approaches to the problem of face recognition have also been introduced. We have combined the classical LBP with logarithmic image processing and the human visual system and introduced PLIP-LBP feature extractors. Computer simulations have shown that all these modified LBP operators can improve the rate of recognition. For example recognition rates as high as 99%, 97% and 98.5% were obtained using logarithmic LBP, PLIP-LBP in logarithmic domain and HVS based LBP respectively as compared to 96.5% using the classical LBP operator using the AT&T Laboratories face database. We have also varied the number and configuration of images used for registration and obtained improved average rates of recognition using the modified LBP operators. Two types of classifiers were used for our experiments and we have found that the modified LBP operators result in improved recognition rates in either case. This

shows the robustness of these operators. We have used rectangular regions of the same size; however regions of different sizes and shapes can also be used. The simplicity of these operators and their robustness against variations in facial expression, aging, illumination and alignment make them suitable for real time applications. We also anticipate that the developed methodology can be extended to several other object detection and recognition tasks. Thus in summary the contributions made in this thesis are

- Introduction of generic framework for HVS based object detection and recognition
- Development of eye detection system
- Introduction of novel logarithmic LBP feature descriptors
- Development of face recognition system based on logarithmic LBP and HVS LBP feature descriptors

Experiments have shown that comparable performance can be obtained using the log-LBP and the HVS-LBP feature descriptors. We have also shown that uniform LBP along with the chi-square distance statistic yields the best performance among all the LBP variants used.

FUTURE WORK

In this thesis we have introduced a novel framework for HVS based object detection and recognition that also incorporates the logarithmic framework. This framework can be extended for any system to increase the robustness.

We have developed two systems based on this generic framework. For eye detection we showed how we can use HVS based image decomposition to improve the rate of detection especially from images that have non-uniform illumination and shadows. While doing feature fusion using logical addition, we placed equal weights to all the HVS based regions. Theoretically we can associate different weights to each of the regions before doing the logical fusion of the feature vectors. From our experiments we have seen that the De-Vries Rose and the Saturation regions of HVS do not contain much useful information. Hence while doing feature fusion, lower weights can be associated with these regions as compared to the Weber region. Thus we can develop a weighted HVS based eye detection system.

The second system that we developed as part of this thesis is a face recognition system based on logarithmic local binary patterns. Three approaches have been introduced for feature extraction from images. Recognition rates have been considerably improved using the logarithmic LBP and HVS-LBP approach. Hence theoretically we can fuse these different modified LBP feature vectors towards the development of a more robust system. One of the difficulties that arise from fusion of multiple feature vectors is that the dimensionality of the feature vector increases. Hence we also propose to apply a dimensionality reduction technique after doing the feature fusion.

A future endeavor would be to integrate the eye detection and the facial recognition systems together. Specifically, the eye detection can be added as an additional validation step after the

performance of face recognition. Using our system for eye detection, the eyes can be extracted from the facial images and a further validation step can be performed using the eyes recognition.

In [88], sparse signal representation has been used to address the problem of face recognition. It has been shown that if the sparsity in the recognition problem is properly harnessed, the choice of the features is no longer critical provided the number of features is sufficiently large and the sparse representation is correctly computed. It is shown that under this framework, the choice of the feature space is no longer critical because even random features can be used to recover sparse representation and hence can be used to classify test images. A comparison of the proposed approach with this approach can be done and a voting scheme can be developed using recognitions from both of these methods.

APPENDIX A

Publications

- Debashree Mandal, Karen Panetta and Sos Agaian, "*Human Visual System Inspired Object Detection and Recognition*", IEEE International Conference on Technologies for Practical Robot Applications (TePRA), April 23-24, 2012
- Debashree Mandal, Karen Panetta and Sos Agaian, "*Face Recognition Based on Logarithmic Local Binary Patterns*", IS&T/SPIE Electronic Imaging 2013: Image Processing: Algorithms and Systems XI (Conference EI109), February 3 – 7, 2013 (Submitted)

REFERENCES

[1] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*, 2 ed.: CL-Engineering, 2007.

[2] M. Jourlin and J. C. Pinoli, "A Model for Logarithmic Image Processing," *Journal of Microscopy,* vol. 149, pp. 21-35, 1988.

[3] T. G. S. Jr., "Image processing in the context of a visual model," *Proceedings of the IEEE,* vol. 60, pp. 828-842, 1972.

[4] J.-C. Pinoli and J. Debayle, "Logarithmic Adaptive Neighborhood Image Processing (LANIP): Introduction, Connections to Human Brightness Perception, and Application Issues," *EURASIP Journal on Advances in Signal Processing, Hindawi Publishing Corporation,* vol. 2007, August 2006.

[5] A. K. Jain, "Advances in mathematical models for image processing," *Proceedings of the IEEE,* vol. 69, pp. 502-528, 1981.

[6] D. J. Granrath, "The role of human visual models in image processing," *Proceedings of the IEEE,* vol. 69, pp. 552-561, 1981.

[7] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* : San Francisco, CA: W.H. Freeman and Company., 1982.

[8] W. F. Schreiber, *Fundamentals of Electronic Imaging Systems: Some Aspects of Image Processing, 2nd edition*. Berlin, Germany: Springer, 1991.

[9] J.-C. Pinoli, "A general comparative study of the multiplicative homomorphic, log-ratio and logarithmic image processing approaches," *Signal Processing,* vol. 58, pp. 11-45, 1997.

[10] K. Panetta, E. J. Wharton, and S. Agaian, "Parameterization of logarithmic image processing models," *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans,* 2007.

[11] G. T. Fechner, *Elements of Psychophysics.* vol. 1: Holt, Rinehart & Winston, New York, NY, USA, English translation by H. E. Adler., 1960.

[12] L. W. Cahill and G. Deng, "An Overview of Logarithm-Based Image Processing Techniques for Biomedical Applications," *IEEE Int. Conf. Digital Signal Processing,* pp. 93-96, 1997.

[13] G. X. Ritter and J. N. Wilson, "The Image Algebra in a Nutshell," *in Proc. 1987 IEEE Int. Conf. Computer Vision, London,* pp. 641-645, 1987.

[14] J. N. Wilson, G. R. Fischer, and G. X. Ritter, "Implementation and Use of an Image Processing Algebra for Programming Massively Parallel Machines," *in Proc. 1988 IEEE Symp. Frontiers of Massively Parallel Computation, Fairfax, Va,* pp. 587-594, 1988.

[15] I. G. Angus, "Image Algebra: An Object Oriented Approach to Transparently Concurrent Image Processing," *in Proc. 1992 IEEE Scalable High Performance Computing Conf., Seattle, WA,* pp. 9-13, 1992.

[16] G. Buchsbaum, "An Analytical Derivation of Visual Nonlinearity," *IEEE Transactions on Biomedical Engineering,* vol. BME-27, May 1980.

[17] M. Shah, "Guest introduction: the changing shape of computer vision in the twenty-first century," *International Journal of Computer Vision,* vol. 50, pp. 103-110, 2002.

[18] E. H. Weber, "Der tastsinn und das gemeingef¨uhl," *in Handw¨orterbuch der Physiologie, E. Wagner, Ed., vol. 3, pp. 481–588, Friedrich Vieweg & Sohn, Braunschweig, Germany,* 1846.

[19] J. Shen, "On the foundations of vision modeling. I. Weber's law and Weberized TV restoration," *Physica D: Nonlinear Phenomena,* vol. 175, pp. 103-110, 2003.
[20] J.-C. PINOLI, "The Logarithmic Image Processing Model: Connections with Human Brightness Perception and Contrast Estimators," *Journal of Mathematical Imaging and Vision,* vol. 7, pp. 341-358, 1997.
[21] G. Deng and L. W. Cahill, "The logarithmic image processing model and its applications," *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on* pp. 1047-1051, Nov 1993.
[22] M. Jourlin and J. C. Pinoli, "Contrast Definition and Contour Detection for Logarithmic Images," *Journal of Microscopy,* vol. 156, pp. 33-40, 1989.
[23] M. Jourlin and J. C. Pinoli, "Logarithmic Image Processing; The Mathematical and Physical Framework for the Representation and Processing of Transmitted Images," *Advances in Imaging and Electron Physics,* vol. 115, pp. 129-196, 2001.
[24] E. Zaharescu, "Medical Image Enhancement using Logarithmic image Processing," *in Proc. 2005 IASTED Int. Conf. Visualization, Imaging, and Image Processing, Benidorm, Spain, 2005, paper #480-031,* 2005.
[25] S. Agaian, K. Panetta, and A. M. Grigoryan, "A New Measure of Image Enhancement," *in Proc. IASTED 2000 Int. Conf. Signal Processing & Communication, Marbella, Spain,* 2000.
[26] O. R. Vincent and O. Folorunso, "A Descriptive Algorithm for Sobel Image Edge Detection," *Proceedings of Informing Science & IT Education Conference (InSITE),* 2009.
[27] A. K. Jain, *Fundamentals of Digital Image Processing,* 1 ed.: Prentice Hall, 1988.
[28] R. C. González and R. E. Woods, *Digital image processing,* 3 ed.: Prentice Hall, 2002.
[29] J. M. Prewitt, B. S. Lipkin, and A. Rosenfeld, "Object enhancement and extraction," *Picture Processing and Psychopictoris,* 1970.
[30] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 8, pp. 679-698, 1986.
[31] P. Bao, L. Zhang, and X. Wu, "Canny Edge Detection Enhancement by Scale Multiplication," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 27, pp. 1485-1490, September 2005.
[32] N. OTSU, "A Tlreshold Selection Method from Gray-Level Histograms," *IEEE TRANSACTIONS ON SYSTREMS, MAN, AND CYBERNETICS,,* vol. SMC-9, January 1979.
[33] Y.-K. Huo, G. Wei, Y.-D. Zhang, and L.-N. Wu, "An Adaptive Threshold for the Canny Operator of Edge Detection," *Image Analysis and Signal Processing (IASP), 2010 International Conference on* pp. 371-374, April 2010.
[34] N. Nain, G. Jindal, A. Garg, and A. Jain, "Dynamic Thresholding Based Edge Detection," *Proceedings of the World Congress on Engineering,* vol. 1, July 2008.
[35] M. Kaushal, A. Singh, and B. Singh, "Adaptive Thresholding for Edge Detection in Gray Scale Images," *International Journal of Engineering Science and Technology,* vol. 2(6), pp. 2077-2082, 2010.
[36] L. Er-sen, Z. Shu-long, Z. Bao-shan, Z. Yong, X. Chao-gui, and S. Li-hua, "An Adaptive Edge-Detection Method Based on the Canny Operator," *Environmental Science and Information Application Technology, 2009. ESIAT 2009. International Conference on* vol. 1, pp. 465-469, July 2009.

[37] J. Zeng and D. Li, "An Adaptive Canny Edge Detector using Histogram Concavity Analysis," *International Journal of Digital Content Technology and its Applications,* vol. 5, June 2011.
[38] M. K. Kundu and S. K. Pal, "Thresholding for edge detection using human psychovisual phenomena," *Pattern Recognition Letters 4,* pp. 433-441, December 1986.
[39] T.-H. Yu, "Nonlinear gradient-based edge detection algorithms in the telesign system," *Circuits and Systems, 1994. APCCAS '94., 1994 IEEE Asia-Pacific Conference on,* pp. 661-666, 1994.
[40] S. Ameer and O. Basir, "Modifying weber fraction law to postprocessing and edge detection applications," *Communications, Control and Signal Processing, 2008. ISCCSP 2008. 3rd International Symposium on ,,* vol. 871-875, March 2008.
[41] K. Panetta, E. Wharton, and S. Agaian, "Human Visual System-Based Image Enhancement and Logarithmic Contrast Measure," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 38, pp. 174-188, Feb 2008.
[42] E. Wharton, K. Panetta, and S. Agaian, "Human Visual System Based Multi-Histogram Equalization for Non-Uniform Illumination and Shoadow Correction," *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on* vol. 1, pp. I-729-I-732, April 2007.
[43] A. Almuntashri and S. Agaian, "Human visual system-based edge detection using image contrast enhancement and logarithmic ratio " *2010 Proc. of SPIE,* vol. 7708, 2010.
[44] E. Wharton, S. Agaian, and K. Panetta, "A logarithmic measure of image enhancement," *Proc. SPIE 6250, 62500P (2006),* 2006.
[45] Y. Suzuki and T. Shibata, "An Edge-Based Face Detection Algorithm Robust Against Illumination, Focus, And Scale Variations," *European Signal Processing Conference EUSIPCO 2004, Vienna, Austria,* vol. , September 2004.
[46] Y. Suzuki and T. Shibata, "Multiple-resolution edge-based feature representations for robust face segmentation and verification," *Proc. of Euro. Sig. Proc. Conf., Antalya,* September 2005.
[47] S. Veeraraghavan, K. Panetta, and S. Agaian, "Web based integrated framework for security applications," *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on* pp. 1889-1895, October 2010.
[48] Y. Suzuk and T. Shibata, "A Directional-Edge-Based Face Detection Algorithm Adapted to the VLSI Image Recognition System," in *Image Fusion and Its Applications*, Y. Zheng, Ed., ed Alcorn State University, USA: InTech, June 2011.
[49] Y. Suzuki and T. Shibata, "Multiple-clue face detection algorithm using edge-based feature vectors," *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on,* vol. 5, pp. V- 737-40, May 2004.
[50] M. Yagi, T. Shibata, C. Tanikawa, and K. Takada, "A Robust Medical Image Recognition System Employing Edge-Based Feature Vector Representation." vol. 2749, J. Bigun and T. Gustavsson, Eds., ed: Springer Berlin / Heidelberg, 2003, pp. 727-728.
[51] A. Kingston, S. Colosimo, P. Campisi, and F. Autrusseau, "Lossless Image Compression and Selective Encryption using a Discrete Radon Transform," *Image Processing, 2007. ICIP 2007. IEEE International Conference on* vol. 4, pp. IV-465-IV-468, Sept. 16 2007-Oct. 19 2007.

[52] S. Nercessian, K. Panetta, and S. Agaian, "Improving edge-based feature extraction using feature fusion," *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on* pp. 679-684, 12-15 Oct. 2008.

[53] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on,* vol. 1, pp. 886-893 June 2005.

[54] O. Ludwig, D. Delgado, V. Goncalves, and U. Nunes, "Trainable classifier-fusion schemes: An application to pedestrian detection," *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on,* pp. 1-6, Oct 2009.

[55] Y. Fu, L. Cao, G. Guo, and T. S. Huang, "Multiple feature fusion by subspace learning," *Proceedings of the 2008 international conference on Content-based image and video retrieval,* pp. 127-134, 2008.

[56] S. Yan, H. Wang, X. Tang, and T. Huang, "Exploring Feature Descriptors for Face Recognition," *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on* vol. 1, pp. I-629-I-632, April 2007.

[57] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang, "Large-scale image classification: Fast feature extraction and SVM training," *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on,* pp. 1689-1696, June 2011.

[58] W. T. Freeman and M. Roth, "Orientation histograms for hand gesture recognition," *Intl. Workshop on Automatic Faceand Gesture- Recognition, IEEE Computer Society, Zurich, Switzerland,* pp. 296-301, June 1995.

[59] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 24, pp. 971-987, July 2002.

[60] D. Maturana, D. Mery, and Á. Soto, "Face Recognition with Local Binary Patterns, Spatial Pyramid Histograms and Naive Bayes Nearest Neighbor Classification," *Chilean Computer Science Society (SCCC), 2009 International Conference of the,* pp. 125-132, Nov 2009.

[61] T. Ahonen, A. Hadid, and M. Pietikainen, "Face Recognition with Local Binary Patterns," *Computer Vision - ECCV 2004,* pp. 469-481, 2004.

[62] R. C. González, R. E. Woods, and S. L. Eddins, *Digital Image processing using MATLAB,* 2 ed.: Pearson Prentice Hall, 2004.

[63] M. Hangarge and B. V. Dhandra, "Shape and Morphological Transformation Based Features for Language Identification in Indian Document Images," *Emerging Trends in Engineering and Technology, 2008. ICETET '08. First International Conference on* pp. 1175-1180, July 2008.

[64] T. C. Bockholt, G. D. C. Cavalcanti, and C. A. B. Mello, "Document image retrieval with morphology-based segmentation and features combination," *Proc. SPIE 7874, 787415* 2011.

[65] M. Shafi and P. W. H. Chung, "Eyes extraction from facial images using edge density," *Cybernetic Intelligent Systems, 2008. CIS 2008. 7th IEEE International Conference on,* pp. 1-6, Sept 2008.

[66] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on,* pp. 586-591, June 1991.

[67] V. P. Kshirsagar, M. R. Baviskar, and M. E. Gaikwad, "Face recognition using Eigenfaces," *Computer Research and Development (ICCRD), 2011 3rd International Conference on* vol. 2, pp. 302-306, March 2011.
[68] M. Agarwal, H. Agrawal, N. Jain, and M. Kumar, "Face Recognition Using Principle Component Analysis, Eigenface and Neural Network," *Signal Acquisition and Processing, 2010. ICSAP '10. International Conference on* pp. 310-314, Feb 2010.
[69] N. Mittal and E. Walia, "Face Recognition Using Improved Fast PCA Algorithm," *Image and Signal Processing, 2008. CISP '08. Congress on,* vol. 1, pp. 554-558, May 2008.
[70] H. F. Liau, L.-M. Ang, and K. P. Seng, "A multiview face recognition system based on eigenface method," *Information Technology, 2008. ITSim 2008. International Symposium on,* vol. 2, pp. 1-5, Aug 2008.
[71] M. Eriksson and N. P. Papanikotopoulos, "Eye-tracking for detection of driver fatigue," *Intelligent Transportation System, 1997. ITSC '97., IEEE Conference on,* pp. 314-319, Nov 1997.
[72] X. Deng, C.-H. Chang, and E. Brandle, "A new method for eye extraction from facial image," *Electronic Design, Test and Applications, 2004. DELTA 2004. Second IEEE International Workshop on,* pp. 29-34, Jan 2004.
[73] M. Rizon and T. Kawaguchi, "Automatic eye detection using intensity and edge information," *TENCON 2000. Proceedings* vol. 2, pp. 415-420, 2000.
[74] M. Betke and W. J. Mullally, "Preliminary investigation of real-time monitoring of a driver in city traffic," *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE* pp. 563-568, 2000.
[75] A. L. Yuille, D. S. Cohen, and P. W. Hallinan, "Feature extraction from faces using deformable templates," *Computer Vision and Pattern Recognition, 1989. Proceedings CVPR '89., IEEE Computer Society Conference on,* pp. 104-109, Jun 1989.
[76] T. Kawaguchi, D. Hidaka, and M. Rizon, "Detection of eyes from human faces by Hough transform and separability filter," *Image Processing, 2000. Proceedings. 2000 International Conference on,* vol. 1, pp. 49-52, 2000.
[77] T. Ahonen, A. Hadid, and M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 28, pp. 2037-2041, 2006.
[78] C. H. CHAN, "Multi-scale Local Binary Pattern Histogram for Face Recognition," Doctor of Philosophy, Centre for Vision, Speech and Signal Processing, School of Electronics and Physical Sciences, University of Surrey, Guildford, Surrey, U.K, September 2008.
[79] B. Da and N. Sang, "Local binary pattern based face recognition by estimation of facial distinctive information distribution," *Opt. Eng. 48, 117203 (Nov 04, 2009),* 2009.
[80] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face Recognition: A Literature Survey," *ACM Computing Surveys,* vol. 35, pp. 399–458, 2003.
[81] K. Etemad and R. Chellappa, "Discriminant analysis for recognition of human face images," *Journal of the Optical Society of America,* vol. 14, pp. 1724-1733, 1997.
[82] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 19, pp. 711-720, July 1997.

[83] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on* pp. 84-91, 1994.

[84] 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.

[85] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* vol. 19, pp. 775-779, 1997.

[86] B. Heisele, P. Ho, J. Wu, and T. Poggio, "Face recognition: component-based versus global approaches," *Computer Vision and Image Understanding,* vol. 91, pp. 6-21, 2003.

[87] C. Shan, S. Gong, and P. W. McOwan, "Robust facial expression recognition using local binary patterns," *Image Processing, 2005. ICIP 2005. IEEE International Conference on,* vol. 2, pp. II- 370-3, 2005.

[88] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust Face Recognition via Sparse Representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 31, pp. 210-227, Feb 2009.