Visual recognition by subspace approaches on LBP features

Ren, Jianfeng

2015

Ren, J. (2015). Visual recognition by subspace approaches on LBP features. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/62538>

<https://doi.org/10.32657/10356/62538>

Visual Recognition by Subspace Approaches on LBP Features

REN Jianfeng

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2015

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

.....

Date

.....

REN Jianfeng

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor, Associate Professor JIANG Xudong, for his guidance, patience and encouragement. His inspiring nature and continuous encouragement have been great source of motivation throughout my doctoral study. His sharp insights, excellent guidance and strict attitude towards research have been a model for me and will continuously benefit me throughout my lifetime.

I also would like to take this opportunity to express my sincere gratitude to my co-advisor Assistant Professor YUAN Junsong. His creative research ideas, insightful advice and warmly encouragement greatly help my research.

I sincerely thank my friends Dr. YANG Huijuan, Dr. CAO Hong, Dr. GENG Cong, Dr. LIU Siyuan, Dr. CHEN Changsheng, Dr. LI Sheng, Dr. FAN Jiayuan, Dr. Amit SATPATHY, Dr. MIAO Zhenwei, Dr. XIAO Yang, Dr. FRANK GUAN Yunqing, Mr. LIANG Hui, Mr. LAI Jian, Mr. XU Di, Mr. LI Bingbing, Mr. Zhao Mengyao, Mr. GUO Yu, Mr. TAN Fuwen, Mr. CHEN Chongyu and Mr. HOU Junhui for their encouragements and helps. I am grateful to Stewart CHU for his technical support.

I sincerely thank the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, for offering me the opportunity of doctoral study. I also sincerely thank my co-advisor, Assistant Professor YUAN Junsong, together with BeingThere Centre, Institute of Media Innovation, Nanyang Technological University, Singapore, for offering me the research-associate position so that I have financial support to complete my doctoral study.

Finally, I would like to dedicate this work to my wife, my son, my daughter, my parents, my parents-in-law and my sister's family for their persistent love, continuous encouragement and endless support.

Abstract

Subspace approaches have been widely used in visual recognition. Traditionally, they are applied on holistic features derived by vectorizing raw-image pixels. However, the performance is often limited by rigorous image alignment and high computational cost. Recently, local binary pattern (LBP) becomes popular because of robustness to illumination variations and alignment error, and fast and easy feature extraction. However, it still has some limitations such as sensitivity to image noise, high feature dimensionality, unreliable estimation of LBP histogram and non-Euclidean feature space. Subspace approaches have been applied on LBP feature and partially resolved the problems. However, non-Gaussian distribution of LBP feature limits the performance gain brought by subspace approaches.

In this thesis, we exploit the advantages of both subspace approaches and LBP feature. In Chapter 3, we enhance the performance of subspace approaches on holistic features, i.e. we propose a fast and accurate subspace face/eye detector. Then, we boost the performance of LBP feature by a noise-resistant LBP with an embedded error-correction mechanism in Chapter 4, and LBP structure optimization in Chapter 5 that better captures image characteristic. Finally, we improve the performance of subspace approaches on LBP features by two learning-based LBP features in Chapter 6, and a Chi-squared transformation in Chapter 7 that tackles the non-Gaussian distribution of LBP feature. These contributions are discussed in detail as follow.

Firstly, we improve the subspace approaches on holistic features. We propose a natural and non-intrusive way to secure mobile devices, i.e. a complete and fully automated face verification

system. The proposed subspace face/eye detector locates the eyes at a much higher precision than Adaboost face/eye detector at a comparable speed thanks to discriminative APCDA features and attentional cascade strategy. The proposed approach that determines the class-specific threshold without sacrificing training data for validation data further boosts the performance. The proposed system is systematically evaluated on O2FN, AR and CAS-PEAL databases, and achieves a better recognition performance at a comparable speed compared with other approaches.

Secondly, we improve the noise robustness of LBP feature. LBP feature is sensitive to noise. Local ternary pattern (LTP) partially solves the problem. However, both LBP and LTP treat the corrupted patterns as they are. Thus, we propose a noise-resistant LBP to preserve the image local structures in presence of noise. We encode the small pixel difference as an *uncertain* bit first, and then assign its value to form possible uniform codes that represent image micro-structures. In addition, we find that some image patterns such as lines are not captured in uniform codes. We then propose an extended noise-resistant LBP to capture line patterns. Both proposed approaches are shown more resistant to noise than other LBP variants.

Thirdly, we propose to optimize the LBP structure to better capture the image characteristic. LBP features using more bits extracted in a large structure often improve the performance, at the cost of high dimensionality. We propose to treat binarized pixel differences between a pixel and its neighbors as features, and select an optimal subset to compose the structure. As such features are strongly correlated, we propose an incremental Maximal-Conditional-Mutual-Information (MCMI) scheme for LBP structure optimization. The proposed approach has shown a superior performance over the state-of-the-arts results on many applications.

Fourthly, we propose two learning-based LBP features by utilizing subspace approaches, which better remove the noise in LBP feature due to insufficient pixels and unbalance nature of LBP histogram. For uniform LBP, unreliable non-uniform patterns are grouped into one bin to suppress noise. However, useful or noisy features are different for each task. Thus,

two learning-based approaches are proposed to remove noise in LBP feature. In addition, a “super histogram” is proposed for dynamic texture recognition to improve feature reliability. The proposed approaches achieve a superior performance on face, dynamic texture and scene recognition.

Lastly, we propose a Chi-squared transformation for subspace approaches on LBP feature. Subspace approaches utilizing up to the second-order statistics are optimal under Gaussian assumption. However, due to non-negative and simplex constraints, LBP feature deviates significantly from Gaussian. We propose a Chi-squared transformation so that the resulting feature fits better to Gaussian. The proposed transform leads to a two-class asymmetric classification problem, and hence we propose a CST-APCA approach to tackle the asymmetric problem. Extensive experiments on various recognition tasks demonstrates the effectiveness of the proposed transformation.

The contributions of this thesis are summarized as follow:

- A complete face verification system is built on mobile devices, in which the proposed face/eye detector aligns the face more precisely at a comparable speed, and the proposed class-specific threshold further boosts the performance.
- We propose a noise-resistant LBP to recover the corrupted image structures, and an extended noise-resistant LBP to capture line patterns.
- We propose to optimize LBP structure to better capture image characteristics, and a MCMI scheme to better handle strong correlation of binarized-pixel-difference features.
- We propose two learning-based approaches to better remove the noise of LBP features, and a “super histogram” to improve its reliability for dynamic texture recognition.
- We propose a Chi-squared transformation to handle non-Gaussian distribution of LBP feature, and a CST-APCA approach to tackle the asymmetric nature of our formulation.

Contents

Acknowledgements	i
Abstract	iii
List of Figures	xiv
List of Tables	xxi
List of Acronyms	xxiv
1 Introduction	1
1.1 Motivation	1
1.2 Contributions	6
1.3 Organization of the Thesis	7
2 Literature Review	9
2.1 Linear Subspace Approaches	9
2.1.1 Eigenface	11
2.1.2 Asymmetric PCA	12

2.1.3	Fisherface	13
2.1.4	Null-Subspace Approach	14
2.1.5	Dual-Space LDA	14
2.1.6	Bayesian Maximum Likelihood	16
2.1.7	ERE Approach	18
2.1.8	Summary and Discussion	19
2.2	Local Binary Pattern	20
2.2.1	LBP Variants Robust to Image Noise	22
2.2.1.1	Uniform LBP	22
2.2.1.2	Fuzzy LBP	23
2.2.1.3	LTP and Its variants	24
2.2.1.4	Preprocessing	26
2.2.1.5	Summary and Discussion	27
2.2.2	Reducing Feature Dimensionality of LBP	27
2.2.2.1	Direct-Bin Selection/Extraction	28
2.2.2.2	Handcrafted LBP Structures	29
2.2.2.3	LBP Structure Learning	31
2.2.2.4	Summary and Discussion	31
2.3	Subspace Approaches on LBP Features	32
2.3.1	Principal Component Analysis on LBP Features	32
2.3.2	Linear Discriminative Analysis on LBP Feature	33

2.3.3	Kernel PCA and Laplacian PCA on LBP Features	34
2.3.4	Dimensionality Reduction by Partial Least Squares	35
2.3.5	Locality Preserving Projections on LBP Feature	36
2.4	Summary	37
3	A Complete and Fully Automated Face Verification System on Mobile Devices	39
3.1	Introduction	39
3.2	Related Work	42
3.3	The Proposed Face Verification System on Mobile Devices	44
3.3.1	Face Detection	44
3.3.2	Eye Detection	48
3.3.3	Face Verification	49
3.4	O2FN Mobile Face Database Creation	54
3.5	Experimental Results	56
3.5.1	Face Detection Results on the O2FN Database	56
3.5.2	Eye Detection Results on the O2FN Database	59
3.5.3	Face Verification Results on the O2FN Database	62
3.5.4	System-Level Comparison on the O2FN Database	65
3.5.5	Experimental Results on the AR Database	66
3.5.6	Experimental Results on the CAS-PEAL Database	67
3.5.7	Implementation on Mobile Devices	68
3.6	Summary	69

4	Noise-Resistant Local Binary Pattern with an Embedded Error-Correction Mechanism	71
4.1	Introduction	72
4.2	Related Work	73
4.3	Noise-Resistant LBP	74
4.3.1	Problem Analysis of LBP and LTP	74
4.3.2	Proposed Noise-Resistant LBP	77
4.3.3	Proposed Extended Noise-Resistant LBP	82
4.4	Experimental Results	83
4.4.1	Face Recognition on the AR Database	86
4.4.1.1	Resistant to Additive Gaussian Noise	87
4.4.1.2	Resistant to Additive Uniform Noise	90
4.4.2	Texture Recognition on Outex-13 dataset	91
4.4.3	Face Recognition on the Extended Yale B Database	92
4.4.4	Face Recognition on the O2FN Mobile Database	93
4.4.5	Protein Cellular Classification on 2D Hela Database	95
4.4.6	Image Segmentation	97
4.4.7	Comparison of Computational Complexity	98
4.5	Summary	99
5	Learning LBP Structure by Maximizing the Conditional Mutual Information	101
5.1	Introduction	101

5.2	Related Work	103
5.3	Overview of Proposed Approach	104
5.4	An Incremental Maximal-Conditional-Mutual-Information Scheme for LBP Structure Learning	107
5.4.1	Review of Feature-Selection Algorithms Based on Mutual Information	107
5.4.2	A Close Approximation to $I(\mathbf{x}_m; c)$	110
5.4.3	Proposed Incremental MCMi Scheme	112
5.5	Experimental Results	114
5.5.1	Comparison in Approximating $I(\mathbf{x}_m; c)$	114
5.5.2	Texture Classification on the KTH-TIPS-2a Dataset	116
5.5.3	Scene Recognition on the 21-Land-Use Dataset	117
5.5.4	Scene Recognition on the 8-Event Dataset	119
5.5.5	Dynamic Texture Recognition on the DynTex++ Dataset	121
5.5.6	Dynamic Texture Recognition on the UCLA Database	123
5.5.7	Analysis of Computational Complexity	125
5.6	Summary	126
6	Principal Histogram Analysis of LBP and its Applications in Visual Recognition	128
6.1	Introduction	128
6.2	Related Work	130
6.3	The Proposed Learning-Based LBP	131
6.3.1	The Proposed Patch-Independent Learning-Based LBP	132

6.3.2	The Proposed Patch-dependent Learning-Based LBP	134
6.3.3	Relations to Other Approaches	134
6.4	The Proposed Super Histogram	135
6.5	Experiments on Face Recognition	137
6.5.1	Experiments on the FERET database	137
6.5.2	Experiments on the AR database	138
6.5.3	Summary of Face Recognition Results	140
6.6	Experiments on Dynamic Texture Recognition	141
6.6.1	DT Recognition on the UCLA Database	141
6.6.2	DT Recognition on the DynTex database	143
6.6.3	DT Recognition on the DynTex++ Database	145
6.7	Experiments on Scene Recognition	147
6.8	Summary	150
7	A Chi-Squared-Transformed Subspace of LBP Histogram for Visual Recognition	151
7.1	Introduction	151
7.2	Related Work	153
7.3	The Proposed Approach	153
7.3.1	Modeling the PDF of LBP Histogram	153
7.3.2	Proposed Chi-Squared Transformation	158
7.3.3	Normality Analysis Using Moments	161
7.3.3.1	Skewness and Kurtosis of LBP Histogram	162

7.3.3.2	Skewness and Kurtosis of z After CST	163
7.3.3.3	Jarque-Bera test	164
7.3.4	Chi-Squared-Transformed Subspace Analysis	165
7.4	Experimental Results	168
7.4.1	Jarque-Bera Normality Test	169
7.4.2	Face Recognition on the AR Database	170
7.4.3	Face Recognition on the O2FN Mobile Database	171
7.4.4	Face Recognition on the LFW Database	172
7.4.5	Protein Cellular Classification on the 2D Hela Database	174
7.4.6	Dynamic Texture Recognition on the UCLA Database	176
7.4.7	Dynamic Texture Recognition on the DynTex++ Dataset	178
7.4.8	Computational Complexity Analysis	179
7.4.9	Summary and Discussion	180
7.5	Summary	181
8	Conclusions and Future Work	182
8.1	Contributions on Subspace Approaches on Holistic Features	182
8.2	Contributions on Improvements of LBP Features	183
8.2.1	Noise-Resistant Local Binary Pattern	183
8.2.2	Dimensionality Reduction by Optimizing LBP Structure	184
8.3	Contributions on Subspace Approaches on LBP Features	185
8.3.1	Learn-based Local Binary Patterns	185

8.3.2	A Chi-squared Transformation on LBP Features	185
8.4	Future Work	186
8.4.1	An Improved Noise-Resistant Local Binary Pattern	186
8.4.2	Learning Discriminative Patch-Wise LBP Structures	186
8.4.3	A Chi-squared transformed Data-drive NRLBP	187
8.4.4	Learning Optimal Structure for Local Ordinal Pattern	187
8.4.5	Optimizing Multi-channel LBP Structure	187
8.4.6	Improve the Fitness to Gaussian for Histogram-like Features	188
8.4.7	Subspace Approach on Large Dataset	188
Author's Publications		190
Bibliography		192

List of Figures

1.1	Block diagram of the thesis.	6
3.1	Block diagram of the proposed face verification system on mobile devices. . . .	44
3.2	Block diagram of ERE approach.	52
3.3	Illustration for determining the class-specific threshold based on the training data. The small stars, circles, squares and diamonds represent samples of Class 1-4, respectively. The big ones represent the class centers. The distance of a class center to its nearest class center is used as the indicator of the CST.	54
3.4	Sample images of the O2FN database.	55
3.5	Experimental results on the O2FN database. (a) Eigen-spectrum plot for the eigenvalue λ_k^p, λ_k^q of the face/non-face covariance matrix, and the corresponding variance v_k^p, v_k^q of the face/non-face testing samples. (b) The EER for the PCA/APCA face detector. (c) The EER for the APCDA face detector.	57
3.6	Experimental results on the O2FN database. (a) Eigen-spectrum plot for the eigenvalue λ_k^p, λ_k^q of the eye/non-eye covariance matrix, and the corresponding variance v_k^p, v_k^q of the eye/non-eye testing samples. (b) The comparison of the PCA/APCA eye detector in terms of EER. (c) The EER of the APCDA eye detector.	60

3.7	The sample images of face/eye detection results for the C-APCDA face/eye detector on the O2FN database.	63
3.8	The ROC for ERE approach based on the class-specific threshold	65
4.1	(a) An example of LBP encoding scheme for the smooth region with small image noise. LBP is sensitive to image noise. (b) An example of LTP encoding process. LTP doubles the number of patterns compared with LBP.	76
4.2	Illustration of encoding process of NRLTP and comparison to LBP and LTP. Fig. (a), (b), (c), (d) are corresponding to $m = 1, 2, 3, 4$ resulting NRLBP codes, respectively. Fig. (e) shows an example that no uniform code can be formed. The proposed NRLBP is significantly different from LBP and LTP. Threshold t is chosen as 2 for LTP and NRLBP in this figure.	80
4.3	The histogram of LBP, LTP and NRLBP for the image shown in Fig. 4.7(c). LTP histogram is the concatenation of positive LBP histogram and negative LBP histogram. The last bin of each histogram is corresponding to non-uniform patterns. Compared with LBP histogram and LTP histogram, NRLBP significantly reduces non-uniform patterns from about 35% to about 10%. The proposed NRLBP corrects a large amount of noisy non-uniform patterns back to uniform patterns.	81
4.4	Local primitives detected by $LBP_{8,2}^{u2}$	82
4.5	Samples of line patterns. Those three rows are corresponding to horizontal, diagonal and vertical lines. The diagonal lines are rare patterns for natural images and hence discarded. The remaining horizontal and vertical lines are the proposed extended set of uniform patterns.	83
4.6	48 extended uniform patterns of ENRLBP.	83

4.7	The images with additive Gaussian noise of $\sigma = 0, 0.05, 0.1, 0.15$, respectively.	87
4.8	The recognition rates of LBP, LTP, DLBP, FLBP, NRLBP and ENRLBP using Chi-square distance vs. threshold t on the AR database injected with Gaussian noise $\sigma = 0.05, 0.10, 0.15$. As the noise level increases, the optimal threshold increases.	89
4.9	The images with uniform noise of $p = 0.1, 0.2, 0.4, 0.7$, respectively.	90
4.10	Row 1 shows the sample images of Outex-13 dataset injected with Gaussian noise of $\sigma = 0.05, 0.10, 0.15$ and uniform noise of $p = 0.1, 0.2, 0.4$, respectively. Row 2 shows the respective images after the preprocessing as in [1]. . .	91
4.11	The 1st row and 2nd row show the samples of geometrically normalized and photometrically normalized images for the extended Yale B database, respectively. The leftmost image is the gallery image, and the other 3 images taken under extreme lighting conditions are the probe images.	93
4.12	The samples of geometrically normalized (Row 1) and photometrically normalized (Row 2) face images of the O2FN databases.	94
4.13	Sample images of the 2D Hela database.	96
4.14	Example of segmentation results on a mix-texture image for LBP, LTP, dominant LBP, fuzzy LBP, the proposed NRLBP and ENRLBP.	97
4.15	Example of segmentation results on a natural scene image for LBP, LTP, dominant LBP, fuzzy LBP, the proposed NRLBP and ENRLBP.	98
5.1	Block diagram of the proposed approach. The neighbors of learned LBP structure are marked as “o”.	105

5.2	Potential candidates. (a) The binarized pixel differences between 24 neighbors and center pixel in a neighborhood of 5×5 pixels as potential candidates. LQP $Disc_5^{3*}$ [2] utilizes the same set of neighbors. CENTRIST feature [3] utilizes 8 neighbors highlighted in yellow. (b) The binarized pixel differences between 26 neighbors and center pixel of Frame t as potential candidates for STLBP.	107
5.3	The proposed MCMI achieves a closer approximation to high-order mutual information $I(\mathbf{x}_m; c)$ than MmCMI [4]. The proposed MCMI is a tighter lower bound of $I(\mathbf{x}_m; c)$	115
5.4	The learned LBP structures for the first trial of 21-land-use dataset.	118
5.5	Confusion matrix of the proposed approach with MCMI scheme for the 21-land-use dataset. Only entries with count at least 3 are shown.	119
5.6	The learned LBP structures for the first trial of 8-event dataset.	120
5.7	Confusion matrix of the proposed approach with MCMI scheme for the 8-event dataset. Only entries with count at least 3 are shown.	121
5.8	The learned spatial-temporal LBP structures of the DynTex++ dataset for 5 trials.	122
5.9	The average recognition rate over 5 trials for the proposed approach built using different number of neighbors.	122
6.1	Samples of dynamic textures from DynTex database.	130
6.2	Block diagram of patch-independent learning-based LBP. Covariance matrices of all patches are summed up and one unified projection matrix P is obtained for all patches.	132

6.3	Block diagram of patch-dependent learning-based LBP. In order to capture the intrinsic image characteristic of different patches, one projection matrix is built for each patch.	134
6.4	The super histogram and the histogram of super image.	137
6.5	Sample images of the FERET database	138
6.6	Plot of the recognition rates for PI-LBP and PD-LBP on the FERET database. .	139
6.7	Sample images of the AR database	139
6.8	Plot of the recognition rate for PI-LBP and PD-LBP on the AR database.	140
6.9	Samples of the UCLA database.	142
6.10	Samples of the DynTex database.	144
6.11	The average recognition rates vs. the number of patches per row/column on the DynTex database.	145
6.12	Samples of the DynTex++ database.	146
6.13	The average recognition rates of LBP, ULBP, PCA-cLBP, PI-LBP and PD-LBP vs. N_p on the DynTex++ database.	147
6.14	The recognition rates of the proposed approach and DFS [5] for each class of the DynTex++ database.	148
6.15	Sample images of the 15-class scene database.	149

7.1	Comparison of the distributions of the features before and after the proposed Chi-squared transformation. (a)-(d) Sample distributions of LBP-histogram features $h \in [0, 1]$. They look like a truncated Gaussian distribution. (e)-(h) The distributions of the transformed features $z \in [-1, 1]$, derived by applying the proposed transformation on the features with distributions shown in (a)-(d). They fit better to Gaussian distribution.	154
7.2	Illustration of a truncated Gaussian distribution.	155
7.3	Δ_{σ^2} increases monotonically with α	156
7.4	α_S increases monotonically with α	157
7.5	Plot of truncation measure $ \alpha_S $ of the LBP-histogram features and $ \alpha_z $ of the features after the proposed Chi-squared transformation. For a better view, $ \alpha_S $ is sorted in an ascending order and plotted together with the corresponding $ \alpha_z $. $ \alpha_z $ is consistently much larger than $ \alpha_S $, which indicates that the truncation is less severe after the proposed transformation.	158
7.6	Illustration of the skewness and the kurtosis.	161
7.7	Comparison of the Jarque-Bera test statistic J_t before and J_z after the proposed Chi-squared transformation.	164
7.8	Comparison of the test statistic J_t and J_z for features before and after the proposed Chi-squared transformation on (a) the AR database, (b) the O2FN database, (c) the 2D Hela database. For a better view, J_t is sorted in an ascending order. On all three databases, J_z is consistently and significantly smaller than J_t	170

- 7.9 Sample images of the 2D Hela database. Images (a)-(j) correspond to actin, DNA, endosome, ER, golgia, golgpp, lysosome, microtubules, mitochondria and nucleolus, respectively. 175

List of Tables

3.1	Partition of the O2FN database	56
3.2	Comparisons of APCA, APCDA, C-APCDA face detector and Adaboost face detector on the O2FN database	59
3.3	Eye detection rate on the O2FN database for the C-APCDA eye detector of $\epsilon = 0.5\%$, 1% and Adaboost eye detector on the detected face images under different tolerances T , where T is the tolerance of the detected eye differed from the ground-truth eye location.	61
3.4	Performance comparison of APCA, APCDA and C-APCDA eye detector on the O2FN database	62
3.5	Comparisons of the C-APCDA eye detector with other eye detectors on the O2FN database	63
3.6	Comparisons of various subspace face recognition algorithms at different settings in terms of EER on the O2FN database.	64
3.7	Performance comparison of the proposed face verification system and the system built upon Adaboost face/eye detector, evaluated on 2000 images of the O2FN mobile face database for 5 trials	66

3.8	Performance comparison of the proposed face verification system and the system built upon Adaboost face/eye detector, evaluated on 1050 images of 75 subjects on the AR database	66
3.9	Performance comparison of the proposed face verification system and the system built upon Adaboost face/eye detector, evaluated on 2359 images of 101 subjects on the CAS-PEAL database.	68
4.1	Summary of the approaches compared with, the classifiers used and the application tested on.	84
4.2	Summary of the average recognition rate, the standard derivation, the optimal threshold t for LTP, NRLBP, ENRLBP and the optimal d for FLBP on the AR database injected with Gaussian noise.	88
4.3	Summary of the average recognition rate and the standard derivation of each approach at the optimal setting on the AR database injected with uniform noise.	90
4.4	Texture recognition on the Outex-13 database injected with Gaussian noise and uniform noise.	92
4.5	The face recognition rate and the optimal threshold on the extended Yale B database.	94
4.6	Performance comparison for face recognition on the O2FN database.	95
4.7	The performance comparison for protein cellular classification on the 2D Hela database in terms of recognition rate and time.	96
5.1	Correlation coefficient between $I^t(\mathbf{x}_m; c)$ and $I(\mathbf{x}_m; c)$ for MmCMI and the proposed MCMI scheme.	115

5.2	Comparisons with the other approaches on the KTH-TIPS-2a dataset for texture classification.	117
5.3	Comparisons with the state of the art on the 21-land-use dataset for scene recognition.	118
5.4	Comparison with the state of the art on the 8-event dataset for scene recognition.	120
5.5	Comparison with the state of the art on the DynTex++ DT dataset.	123
5.6	Comparison with the state of the art on the UCLA DT dataset for 50-class, 9-class and SIR settings.	124
5.7	Comparisons of time consumption for CENTRIST, $Disc_5^{3*}$ LQP and the proposed approach on the 21-land-use dataset and the 8-event dataset.	126
6.1	Summary of performance comparisons on face recognition.	140
6.2	The recognition rates vs. the number of frames in the test sequence.	142
6.3	The average recognition rates on the UCLA DT database for 50-class and 9-class classification problems.	143
6.4	Performance comparisons on Alpha, Beta and Gamma datasets of the DynTex database.	145
6.5	The average recognition rates on the DynTex++ DT database.	147
6.6	Performance comparisons on the 15-scene database.	149
7.1	Comparison of error rates for different approaches on the AR database.	171
7.2	Comparison of error rates for different approaches on the O2FN database.	172
7.3	Comparison of error rates for different approaches on the LFW database.	174
7.4	Comparison of error rates for different approaches on the 2D Hela database.	175

7.5	Comparison with the state of the arts on the UCLA DT dataset for 50-class, 9-class and SIR settings.	177
7.6	Comparison with the state of the arts on the DynTex++ DT dataset.	178
7.7	Comparison of time consumption of different approaches on the high-dim LBP features of the LFW database.	179

List of Acronyms

ADA	Asymmetric Discriminant Analysis
APCA	Asymmetric Principal Component Analysis
APCDA	Asymmetric Principal Component and Discriminant Analysis
API	Application Program Interface
BLBP	Bayesian Local Binary Pattern
BML	Bayesian Maximum Likelihood
BRSP	Randomized-Spatial-Partition-based classifier via Boosting
C-APCDA	Cascade Asymmetric Principal Component and Discriminant Analysis
CBP	Centralized Binary Pattern
CDA	Covariance Discriminant Analysis
CENTRIST	CENSus TRansform hISTogram
CPU	Central Processing Unit
CS-LTP	Center-Symmetric Local Ternary Pattern
CST	Chi-Squared Transformation
CST	Class-Specific Threshold
CST-APCA	Chi-Squared-Transform based Asymmetric Principal Component Analysis
CWLD	Color Weber Local Descriptor
DCT	Discrete Cosine Transform
DFD	Discriminant Face Descriptor

DFS	Dynamic Fractal Spectrum
dLBP	direction-coded Local Binary Pattern
DL-PEGASOS	Distance-Learning Primal Estimated sub-GrAdient SOLver for SVM
DLBP	Dominant Local Binary Pattern
DNA	Deoxyribonucleic Acid
DoG	Difference of Gaussian
DSL	Dual-space LDA
DT	Dynamic Texture
EER	Equal Error Rate
ELTP	Extended Local Ternary Pattern
EML	Enhanced Maximum Likelihood
ENRLBP	Extended Noise-Resistant Local Binary Pattern
EPFDA	Ensemble of Piecewise Fisher Discriminant Analysis
ER	Estrogen receptor
ERE	Eigenfeature Regularization & Extraction
FCBF	Fast Correlation-Based Filtering
F-LBP	Fourier Local Binary Pattern
FLBP	Fuzzy Local Binary Pattern
FLS	Filtering, Labeling and Statistic framework
FRR	False Rejection Rate
GT	Global Threshold
HOG	Histogram of Oriented Gradient
KDCV	Kernel Discriminative Common Vectors
LATP	Local Adaptive Ternary Patterns
LBP	Local Binary Pattern
LBP-TOP	Local Binary Pattern of Three Orthogonal Planes

LDA	Linear Discriminant Analysis
LEP	Local Edge Pattern
LGBP	Local Gabor Binary Pattern
LPP	Locality Preserving Projections
LQP	Local Quantized Pattern
LRP	Local Random Projection
LTP	Local Ternary Pattern
NELBP	Novel Extended Local Binary Pattern
NLDA	Null-space Linear Discriminant Analysis
NNC	Nearest-Neighbor Classifier
NRLBP	Noise-Resistant Local Binary Pattern
NTLBP	Noise Tolerant Local Binary Pattern
MAP	Maximum A Posterior
MB-LBP	Multiscale Block Local Binary Pattern
MBP	Median Binary Pattern
MCMI	Maximal-Conditional-Mutual-Information
MG	Modified G-statistic
ML	Maximum Likelihood
MmCMI	Max-Min-Conditional-Mutual-Information
mMDC	Minimum Mahalanobis Distance Classifier
MJMI	Maximal-Joint-Mutual-Information
mRMR	Min-Redundancy and Max-Relevance
PACT	Principal Component Analysis of Census Transform Histogram
PC	Personal Computer
PCA	Principal Component Analysis
PCA-cLBP	Principal Component Analysis on Concatenated Local Binary Pattern

PCA-mMDC	Principal Component Analysis based Minimum Mahalanobis Distance Classifier
PD-LBP	Patch-Dependent learning-based Local Binary Pattern
PDF	Probability Density Function
PHA	Principal Histogram Analysis
PI-LBP	Patch-Independent learning-based Local Binary Pattern
PLS	Partial Least Squares
PMF	Probability Mass Function
PmSVM	Power-mean Support Vector Machine
RAM	Random-Access Memory
RBF	Radial Basis Function
ROC	Receiver Operating Characteristic
RSP	Randomized Spatial Partition
S-LBP	Sematic Local Binary Pattern
SCI	Sparsity Concentration Index
SF	Scale Factor
SFS	Sequential Forward Selection
SIFT	Scale-Invariant Feature Transform
SILTP	Scale-Invariant Local Ternary Pattern
SIR	Shift-invariant Recognition
SPCK	Spatial Pyramid Co-occurrence Kernel
SPCK+	Extended Spatial Pyramid Co-occurrence Kernel
SPCK++	Second Extended Spatial Pyramid Co-occurrence Kernel
STLBP	Spatial-Temporal Local Binary Pattern
SVM	Support Vector Machine
tLBP	binary-value-transition-coded Local Binary Pattern

ULBP	Uniform Local Binary Pattern
UMACE	Unconstrained Minimum Average Correlation Energy
VLBP	Volume Local Binary Pattern
WLD	Weber Local Descriptor

Chapter 1

Introduction

This thesis focuses on improving both subspace approaches and LBP features for visual recognition. The motivations of our research are explained in Section 1.1. The major contributions of this thesis are summarized in Section 1.2. Finally, the organization of this work is described in Section 1.3.

1.1 Motivation

Subspace approaches are often applied on the holistic features constructed by vectorizing the geometric-normalized gray-level images. They are optimal when the underlying distribution is Gaussian. Under Gaussian assumption, the maximum a posterior (MAP) rule is simplified to a minimum Mahalanobis distance classifier (mMDC). Unfortunately, the Mahalanobis distance cannot be reliably evaluated in the high-dimensional space due to limited training samples. Therefore, various subspace approaches are developed to solve this problem, e.g. Eigenface [6], Fisherface [7], null-space approach [8], dual-space LDA [9], Bayesian Maximum Likelihood approaches [10, 11] and ERE approach [12]. However, subspace approaches on holistic features often suffer from the following limitations: high computational complexity due to large matrix multiplications involved and rigorous image alignment required for a good recognition performance.

For object detection task such as face/eye detection, speed is crucial as millions of sliding windows need to be scanned. However, traditional subspace approaches are not suitable due to their high computational cost. Inspired by the success of Adaboost face detector [13], we utilize an attentional cascade structure to build a cascade subspace face/eye detector. Face/eye detection is highly asymmetric, where only a few are face/eye windows, whereas millions are non-face/non-eye windows. In the early stages of the cascade structure, simple classifiers utilizing a few dimensions are used to filter out a large portion of negative samples and to preserve as many positive samples as possible. In the latter stages, classifiers utilizing more dimensions are used to confirm the candidature of face/eye. In such a way, detection speed is greatly improved. This is critical for mobile devices with limited memory and computational power. By utilizing the proposed face/eye detector, we build a face verification system on mobile devices, which consists of three sub-systems: face detection, face alignment and face recognition. For face verification, a threshold is needed to determine whether it is an imposter user or not. Client-specific threshold often shows a better performance than global threshold. The proposed approach to determine the client-specific threshold without sacrificing part of the training data as the validation data further improves the performance. This work is discussed in Chapter 3 in detail.

In traditional subspace approaches, gray-level images are geometrically and photometrically normalized and used as holistic features. Images are often down-sampled so that the resulting dimension is manageable. However, such a down-sampling may result in significant information loss. In addition, rigorous image alignment is often required to achieve a good performance, which may not be feasible in many cases. Other than holistic features, many features were proposed for visual recognition tasks, e.g. SIFT [14], histogram of oriented gradient (HOG) [15] and Local binary pattern (LBP) [16]. LBP is popular because: 1) The exact intensities are discarded, and only the relative intensities with respect to the center are preserved. Thus, LBP is less sensitive to illumination variations. 2) By extracting the histogram of micro-patterns in a

patch, the exact location information is discarded, and only the patch-wise location information is preserved. Thus, LBP is robust to alignment error. 3) LBP features can be extracted efficiently, which enables real-time image analysis. However, LBP still has some limitations. Chapter 4 is dedicated to address its noise-sensitivity problem. In Chapter 5, we propose to reduce its high dimensionality by deriving the optimal structure for a specific application. In Chapter 6, two learning-based LBP features are proposed to remove the unreliable dimensions of LBP features. In Chapter 7, a Chi-squared transformation is proposed to handle the non-Gaussian distribution of LBP features.

Although LBP has gained much popularity, its sensitivity to noise limits its performance [1]. In [16], uniform LBP was proposed to reduce the noise in LBP histogram. Most LBP codes in natural images are uniform patterns [16, 17]. Uniform patterns are statistically significant, and their occurrence probabilities can be reliably estimated. In contrast, non-uniform patterns are statistically insignificant, and hence noise-prone and unreliable. Local ternary pattern (LTP) was proposed in [1] to tackle the image noise in uniform regions, which partially solves the noise-sensitive problem. However, both LBP and LTP lack a mechanism to recover the underlying image patterns corrupted by noise. Thus, we propose a noise-resistant LBP (NRLBP) to address this issue. As the small pixel differences are vulnerable to noise, we encode them as *uncertain* bits. It is known that uniform patterns represent image micro-structures and non-uniform patterns are most likely corrupted by noise. Thus, we assign the values of *uncertain* bits so as to form uniform patterns. Noise may change an uniform pattern into an unstable non-uniform pattern, whereas the proposed NRLBP corrects many distorted non-uniform patterns back to uniform patterns, which can be viewed as an error-correction process. In addition, we find that line patterns represent an important group of local primitives for pattern recognition. Thus, we propose an extension set of uniform patterns corresponding to line patterns, and propose an extended noise-resistant LBP (ENRLBP). Two proposed approaches are discussed in Chapter 4 in detail.

The second problem of LBP is that its dimensionality may be too high as it increases exponentially with the number of LBP bits. In [16], LBP features were extracted using P neighbors uniformly sampled on a circle at the radius of R to the center pixel. Its histogram has 2^P bins. Recent researches [2, 16, 18] show that LBP features of more bits extracted in a larger neighborhood have more potential to capture complex patterns than using the basic ones, but result in a very high feature dimensionality. It is not feasible to enumerate all histogram bins even for a moderate P . Thus, an optimal LBP structure needs to be determined first to reduce the histogram to a reasonable size. LBP-TOP [19], CS-LTP [20] and LQP [2] are initial attempts to directly reduce the size of LBP structure. However, optimality cannot be guaranteed as they are selected heuristically. More importantly, the LBP structure should be task-dependent as the intrinsic image characteristics of different applications, or even different patches of an image are different. In view of these, we propose to derive the optimal LBP structure first, and then use it to generate LBP features. This is particularly important in the scenarios that we have little prior knowledge of a specific application and it is unclear how to define a good handcrafted structure. We cast LBP structure learning as a point-selection problem. The goal is to select an optimal subset of neighbors to form the LBP structure. Point features are strongly correlated, and hence conventional feature-selection methods may not yield a desirable performance. Thus, we propose an incremental Maximal-Conditional-Mutual-Information (MCMI) scheme to optimize the LBP structure. Chapter 5 is dedicated for LBP structure optimization.

The third problem of LBP is its reliability issue, which mainly arises from insufficient pixels to construct the LBP histogram: 1) Patch-based LBP leads to fewer elements in the histogram of each patch. 2) The image is smooth in nature, and hence some patterns rarely appear in the LBP histogram. The occurrence probabilities of those rare patterns can not be reliably estimated. To tackle this issue, two learning-based approaches are proposed. Principal Histogram Analysis (PHA) is applied on the LBP features of each patch to remove the unreliable information. The proposed approaches are different from [21, 22], in which PCA is applied on the concatenated

LBP feature vector. The dimensionality of the concatenated feature vector is so high that its covariance matrix can not be reliably estimated. In contrast, the covariance matrix of the proposed approaches is calculated on a patch-wise basis and of a much smaller size, and hence can be better estimated. For dynamic texture (DT) recognition, we find that it largely relies on spatial information [23] and each frame of DTs is similar. Thus, a “super histogram” is proposed, which is constructed by averaging the LBP histograms of all frames along the temporal direction. The temporal information is partially transferred to the super histogram. This work is discussed in Chapter 6 in detail.

The fourth problem of LBP feature is its non-Gaussian distribution. Subspace approaches are optimal only when the features follow Gaussian distribution [12, 24, 25]. However, the LBP histogram is bounded in the range of $[0, 1]$. In general, the expected value of LBP-histogram bin is close to 0. The lower bound at 0 has a severe truncation effect on the probability density function (PDF) of LBP features so that it largely deviates from Gaussian. To produce a feature fit better to Gaussian distribution, we propose a Chi-squared transformation based on the following observations. The LBP histogram is severely unbalanced. Frequent patterns may easily overwhelm infrequent patterns if Euclidean distance is used. To enhance the discriminative power of infrequent patterns, distance measures other than Euclidean distance such as histogram intersection, Chi-squared distance and G-statistic are often employed [17, 26, 27]. The proposed transformation inherits the advantages of Chi-squared distance, which well handles the unbalanced nature of LBP histogram. More importantly, our analysis and experiments show that the resulting feature fits better to Gaussian distribution. The proposed Chi-squared transformation converts the original classification problem into a two-class one. This problem is highly asymmetric since the number of inter-class sample pairs is much larger than the number of intra-class pairs. To handle this asymmetric nature, we propose to use asymmetric PCA [24, 25] instead of PCA to remove the unreliable dimensions in the transformed feature space. This work is discussed in Chapter 7 in detail.

Figure 1.1: Block diagram of the thesis.

1.2 Contributions

The major contributions of this thesis are summarized in Fig. 1.1. They can be grouped as three aspects: improvement of subspace approaches on holistic features, improvements of LBP features and improvements of subspace approaches on LBP features. These contributions are listed as follow.

- We propose a complete face verification system on mobile device using subspace approaches. The proposed subspace face/eye detector locates the eyes at a much higher precision than Adaboost face/eye detector. By utilizing attentional cascade strategy, its speed is comparable to Adaboost face/eye detector. The proposed class-specific threshold derived solely using training data further enhances the performance.
- A NRLBP is proposed to improve the robustness to noise. By encoding the *uncertain bits* to form uniform patterns, we develop an error-correction mechanism to recover the distorted image patterns. In addition, an extended NRLBP is proposed to capture line patterns. The proposed NRLBP and ENRLBP are shown more robust to noise than LBP and its variants.

- We propose to derive an optimal LBP structure to better capture image characteristics of different applications or even image patches, and then use it to produce a LBP histogram of a reasonable size. A MCMI scheme is proposed to tackle the strong correlation of binarized pixel differences. The proposed approach has demonstrated a superior performance to the state-of-the-art approaches in texture classification, scene recognition and dynamic texture recognition.
- By utilizing Principal Histogram Analysis, two learning-based approaches, i.e. PI-LBP and PD-LBP, are proposed to improve the reliability of LBP features. Particularly for dynamic texture recognition, a “super histogram” is proposed to further improve its reliability. The proposed approaches achieve a superior performance in face, dynamic texture and scene recognition.
- We find that LBP feature deviates far from Gaussian distribution, and hence propose to model its PDF as a truncated Gaussian. We then propose a Chi-squared transformation to improve its fitness to Gaussian so that the subsequent subspace analysis is optimal. In view of asymmetric nature of our formulation, we propose a CST-APCA approach to better remove the unreliable dimensions in the transformed feature space. The proposed approach significantly improves the performance of subspace approaches on LBP features for various recognition tasks.

1.3 Organization of the Thesis

The remainder of this thesis is organized as follows.

1. In Chapter 2, the representative subspace approaches and recent developments of LBP features are reviewed.
2. In Chapter 3, a fast and accurate cascade APCDA face/eye detector is proposed, and

integrated as a complete face verification system on mobile device.

3. In Chapter 4, a noise-resistant LBP (NRLBP) with an embedded error-correction mechanism and the extended NRLBP are presented.
4. In Chapter 5, we propose a MCMI scheme to derive the optimal LBP structure for a specific application towards the objective of reducing the feature dimensionality.
5. In Chapter 6, we propose PI-LBP and PD-LBP to remove the unreliable dimensions in LBP features, and a “super histogram” to boost the performance for dynamic texture recognition.
6. In Chapter 7, a Chi-squared transformation is proposed to enhance the performance of subspace approaches on LBP features.
7. In Chapter 8, we conclude this thesis by highlighting our contributions and discussing possible future work.

Chapter 2

Literature Review

In traditional subspace approaches, raw image intensities are used as holistic features. Recently, other features such as SIFT, LBP, HOG are often extracted from multiple image patches at multiple scales to boost the recognition performance. It may result in a high-dimensional feature vector, which adversely affects the robust classification. Thus, many approaches utilize linear subspace methods [6–12] to remove the unreliable feature dimensions or to derive a compact feature representation. In this chapter, we review popular linear subspace approaches, recent developments of LBP features and those subspace approaches applied on LBP features.

2.1 Linear Subspace Approaches

Subspace approaches utilizing up to the second-order statistics are optimal under Gaussian assumption [24, 25]. For Bayes classification, we assign the feature $\mathbf{h} = [h_1, h_2, \dots, h_d]^T \in \mathbb{R}^d$ to one of the c categories, i.e. ω_i , so that the posteriori probability $p(\omega_i|\mathbf{h})$ is maximized. This maximum a posterior (MAP) rule is a Bayes decision rule with the 0/1 loss function. As $p(\omega_i|\mathbf{h}) = p(\mathbf{h}|\omega_i)p(\omega_i)/p(\mathbf{h})$ and $p(\mathbf{h})$ is not a function of ω_i , it is equivalent to maximize the following discriminant function:

$$g_i(\mathbf{h}) = \ln p(\mathbf{h}|\omega_i) + \ln p(\omega_i). \quad (2.1)$$

The natural logarithm is a monotonically increasing function, and hence does not affect the decision result but will simplify the evaluation if $p(\mathbf{h}|\omega_i)$ is an exponential function.

Further quantitative analysis requires an analytical form of the class-conditional probability function $p(\mathbf{h}|\omega_i)$. It is often assumed to follow multivariate Gaussian distribution because of the following reasons. Firstly, it is the most natural distribution and the sum of many independent random variables, irrespective of their original distribution, obeys Gaussian distribution. Given mean and variance, it has the maximum entropy of all distributions. Secondly, Gaussian mixture model, which is the weighted sum of a number of Gaussian distributions, in theory can closely approximate arbitrarily shaped distribution. Lastly, dimensionality-reduction techniques such as principal component analysis (PCA), linear discriminant analysis (LDA) and many others utilize up to the second-order statistics, and so does the Gaussian distribution. Under the Gaussian assumption,

$$p(\mathbf{h}|\omega_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{h} - \bar{\mathbf{h}}_i)^T \Sigma_i^{-1} (\mathbf{h} - \bar{\mathbf{h}}_i)\right], \quad (2.2)$$

where $\bar{\mathbf{h}}_i$ is the class mean and Σ_i is the class covariance matrix. The discriminant function (2.1) becomes

$$g_i(\mathbf{h}) = -\frac{1}{2}(\mathbf{h} - \bar{\mathbf{h}}_i)^T \Sigma_i^{-1} (\mathbf{h} - \bar{\mathbf{h}}_i) + b_i. \quad (2.3)$$

In practice, b_i is a threshold to control the error rate of class ω_i at a price of others. It is basically a minimum Mahalanobis distance classifier. As the covariance matrix Σ_i of individual class may not be reliably estimated in many applications, a pooled covariance matrix Σ_w called within-class scatter matrix is often used instead in Eqn. (2.3), which is calculated as:

$$\Sigma_w = \frac{1}{N} \sum_{i=1}^c N_i \Sigma_i \quad (2.4)$$

where N_i is the number of samples in class ω_i and N is the total number of samples. The

decision rule defined in Eqn. (2.3) is modified as:

$$g_i(\mathbf{h}) = -\frac{1}{2}(\mathbf{h} - \bar{\mathbf{h}}_i)^T \Sigma_w^{-1}(\mathbf{h} - \bar{\mathbf{h}}_i) + b_i. \quad (2.5)$$

Then, eigen-decomposition is applied on Σ_w , i.e. $\Sigma_w = \Phi_w \Lambda_w \Phi_w^T$. Denote $y_j = \phi_j^{wT} \mathbf{h}$ and $\bar{y}_{i,j} = \phi_j^{wT} \bar{\mathbf{h}}_i$ as the projection of $\mathbf{h}$ and $\bar{\mathbf{h}}_i$ on ϕ_j^w , respectively. Here, ϕ_j^w is j -th eigenvector of Σ_w . The decision rule is further simplified to:

$$g_i(\mathbf{h}) = \sum_{j=1}^d \frac{(y_j - \bar{y}_{i,j})^2}{\lambda_j^w} + b_i, \quad (2.6)$$

where λ_j^w is j -th eigenvalue of Σ_w .

Unfortunately, the Mahalanobis distance cannot be reliably evaluated in the high-dimensional space due to limited training samples, i.e. small eigenvalues λ_j^w are unreliable and many of them are zero. Therefore, many subspace approaches [6–12] have been developed to address this issue.

2.1.1 Eigenface

In Eigenface [6], PCA is applied on the total scatter matrix $\Sigma_t = \Phi_t \Lambda_t \Phi_t^T$. Σ_t is calculated by

$$\Sigma_t = \sum_{i=1}^N (\mathbf{h}_i - \bar{\mathbf{h}})(\mathbf{h}_i - \bar{\mathbf{h}})^T, \quad (2.7)$$

where $\mathbf{h}_i, i = 1, 2, \dots, N$ is a sample and $\bar{\mathbf{h}}$ is the sample mean. PCA keeps m eigenvectors Φ_m corresponding to the largest eigenvalues. All samples are then projected into the subspace spanned by eigenvectors Φ_m , i.e. $\tilde{\mathbf{h}}_i = \Phi_m^T \mathbf{h}_i$, and a minimum-Mahalanobis-distance classifier similar to Eqn. (2.3) is used for classification. It improves the generalization performance on the novel testing data by removing the subspace spanned by eigenvectors corresponding to the unreliable small eigenvalues of Σ_t , which in turn removes the dimensions corresponding to the small eigenvalues of Σ_w so that Eqn. (2.6) can be evaluated reliably.

2.1.2 Asymmetric PCA

PCA is designed to achieve the least-mean-square reconstruction error. As pointed out in [24, 25], for optimal classification the goal is to remove the unreliable dimensions that are harmful for reliable classification. PCA on the total scatter matrix Σ_t does not effectively remove the unreliable dimensions. Take face detection as an example. The total scatter matrix is equivalent to:

$$\Sigma_t = \frac{N_p}{N_p + N_n} \Sigma_p + \frac{N_n}{N_p + N_n} \Sigma_n + \Sigma_b, \quad (2.8)$$

where N_p and N_n are the number of samples for face and non-face class, Σ_p and Σ_n are the covariance matrices for face and non-face class, respectively. Σ_b is the between-class scatter matrix. More non-face samples can be collected than face samples. More training samples often result in a more reliable covariance matrix if they are properly collected. Thus, Σ_n is more reliable than Σ_p . A larger weight should be assigned to unreliable Σ_p in Eqn. (2.8) so that more dimensions characterized by the small variances of this class can be removed. On the other hand, Σ_p represents the intrinsic face variations whereas Σ_n represents a universal background model. The eigenspectrum of Σ_n is flatter than that of Σ_p . Such a flat eigenspectrum may largely deviate from the ensemble variances. Thus, a larger weight should be assigned to Σ_n . To tackle this asymmetric nature, in [24,25] an asymmetric pooled covariance matrix is constructed by

$$\Sigma_\alpha = \alpha \Sigma_p + (1 - \alpha) \Sigma_n + \Sigma_b, \quad (2.9)$$

where α is the weighting factor of Σ_p and $0 \leq \alpha \leq 1$.

To remove the dimensions corresponding to the small eigenvalues of Σ_p and Σ_n , PCA is applied on Σ_α , i.e. $\Sigma_\alpha = \Phi_\alpha \Lambda_\alpha \Phi_\alpha^T$, where Φ_α and Λ_α correspond to the eigenvectors and eigenvalues of Σ_α , respectively. Eigenvectors $\Phi_{\alpha,m}$ corresponding to m largest eigenvalues are selected and $\mathbf{h}$ is projected into the subspace as $\hat{\mathbf{h}} = \Phi_{\alpha,m}^T \mathbf{h}$. In such a way, the unreliable

dimensions of Σ_p and Σ_n can be better removed.

For face detection, the optimal classifier decides a positive sample if

$$(\hat{\mathbf{h}} - \hat{\mathbf{h}}_n)^T \hat{\Sigma}_n^{-1} (\hat{\mathbf{h}} - \hat{\mathbf{h}}_n) - (\hat{\mathbf{h}} - \hat{\mathbf{h}}_p)^T (\beta \hat{\Sigma}_p)^{-1} (\hat{\mathbf{h}} - \hat{\mathbf{h}}_p) > \hat{\theta}, \quad (2.10)$$

where $\hat{\Sigma}_n = \Phi_{\alpha,m}^T \Sigma_n \Phi_{\alpha,m}$, $\hat{\Sigma}_p = \Phi_{\alpha,m}^T \Sigma_p \Phi_{\alpha,m}$, β is a weighting constant and $\hat{\theta}$ is a threshold. $\hat{\mathbf{h}}_n$ and $\hat{\mathbf{h}}_p$ are class centers of non-face and face in the projected subspace, respectively. The optimal α and β are task-dependent.

2.1.3 Fisherface

In Fisherface [7], the unreliable dimensions corresponding to small eigenvalues of the within-class scatter matrix Σ_w is removed so that the Mahalanobis distance can be evaluated reliably in a small PCA+LDA subspace. Basically, it aims to find a subspace that maximizes the following criterion:

$$\mathbf{W}^* = \underset{\mathbf{W}}{\operatorname{argmax}} \frac{|\mathbf{W}^T \Sigma_b \mathbf{W}|}{|\mathbf{W}^T \Sigma_w \mathbf{W}|}, \text{ s.t. } \mathbf{W}^T \mathbf{W} = \mathbf{I}, \quad (2.11)$$

where Σ_w and Σ_b are within-class scatter matrix and between-class scatter matrix, respectively. If Σ_w is full rank, the optimal $\mathbf{W}$ can be found by solving the following generalized eigen-decomposition problem:

$$\Sigma_b \Phi = \Sigma_w \Phi \Lambda, \quad (2.12)$$

where Φ and Λ are the matrices of eigenvectors and eigenvalues, respectively.

However, due to high dimensionality and limited training samples, Σ_w is often singular. Besides, the dimensions corresponding the small unreliable eigenvalues of Σ_w adversely affect reliable classification. Thus, PCA is applied on the total scatter matrix Σ_t and only the dimensions corresponding to the leading eigenvalues are kept in order to solve the singularity problem of Σ_w and remove the unreliable dimensions of Σ_w . After projecting all samples

into this subspace, the most discriminative dimensions are extracted by solving the generalized eigen-decomposition problem similarly defined in Eqn. (2.12).

Similar to Eigenface, Fisherface improves the generalization performance on the novel testing data by removing the dimensions corresponding to the small eigenvalues of Σ_w so that the Mahalanobis distance can be evaluated reliably.

2.1.4 Null-Subspace Approach

In Fisherface [7], the null subspace of Σ_w is removed to solve the singularity problem of Σ_w . Liu et al. [8] find that this null subspace still contains discriminant information. In this subspace, the samples from the same class are projected to the same point, and the samples from different classes may be projected to different points. In fact, in terms of the criterion defined in Eqn. (2.11), the null subspace contains the most discriminant information.

More specifically, the feature space is divided into two subspaces: the principal subspace spanned by the eigenvectors corresponding to non-zero eigenvalues of Σ_w , and the null subspace spanned by the eigenvectors corresponding to zero eigenvalues of Σ_w . Since the eigenvalues of the null subspace are zero, the Mahalanobis distance is not suitable, and hence Euclidean distance is used. The nearest-neighbor classifier is used for classification. In this approach, only the null subspace of Σ_w is utilized. As we will see shortly, both principal and null subspaces of Σ_w can be used for classification.

2.1.5 Dual-Space LDA

In dual-space LDA [9], both principal and null subspaces of Σ_w are utilized for classification. In the principal subspace, the LDA vectors are computed as:

$$\mathbf{W}_P^* = \operatorname{argmax}_{\mathbf{W}_P} \frac{|\mathbf{W}_P^T \Sigma_b \mathbf{W}_P|}{|\mathbf{W}_P^T \Sigma_w \mathbf{W}_P|}, \text{ s.t. } \mathbf{W}_P^T \Sigma_w \mathbf{W}_P \neq 0, \mathbf{W}_P^T \mathbf{W}_P = 1. \quad (2.13)$$

In the null subspace, the LDA vectors are computed as:

$$\mathbf{W}_C^* = \underset{\mathbf{W}_C}{\operatorname{argmax}} |\mathbf{W}_C^T \Sigma_b \mathbf{W}_C|, \text{ s.t. } \mathbf{W}_C^T \Sigma_w \mathbf{W}_C = 0, \mathbf{W}_C^T \mathbf{W}_C = 1. \quad (2.14)$$

The detailed algorithm is given below:

1. Apply PCA on Σ_w to compute the principal subspace $\mathbb{F}$ and the null subspace $\bar{\mathbb{F}}$, where $\Phi = [\phi_1, \dots, \phi_K]$ and Λ are matrices of eigenvectors and eigenvalues of $\mathbb{F}$. K is the dimensionality of the principal subspace. Estimate the average eigenvalue ρ in $\bar{\mathbb{F}}$ as:

$$\rho = \frac{1}{N - K} \sum_{i=K+1}^N \lambda_i^*, \quad (2.15)$$

where N is the feature dimensionality. The unknown λ_i^* in $\bar{\mathbb{F}}$ are estimated by fitting a nonlinear function to the available portion of the eigenvalue spectrum in $\mathbb{F}$.

2. Σ_b is transformed into $\mathbb{F}$ as:

$$\mathbf{K}_b^P = \Lambda^{-0.5} \Phi^T \Sigma_b \Phi \Lambda^{-0.5}. \quad (2.16)$$

Apply PCA on $\mathbf{K}_b^P$ to compute eigenvectors Ψ_P corresponding to leading eigenvalues.

The projection matrix in the principal space $\mathbb{F}$ is defined as:

$$\mathbf{W}_P = \Phi \Lambda^{-0.5} \Psi_P. \quad (2.17)$$

3. Project all the class centers to $\bar{\mathbb{F}}$ and compute the reconstruction error as:

$$\mathbf{A}_r = \mathbf{A} - \Phi \Phi^T \mathbf{A}, \quad (2.18)$$

where $\mathbf{A}$ is the class-center matrix and $\mathbf{A}_r$ is the projection of $\mathbf{A}$ into $\bar{\mathbb{F}}$. In $\bar{\mathbb{F}}$, Σ_b is transformed to:

$$\mathbf{K}_b^C = (\mathbf{I} - \Phi \Phi^T) \Sigma_b (\mathbf{I} - \Phi \Phi^T). \quad (2.19)$$

Apply PCA on $\mathbf{K}_b^C$ to compute eigenvectors Ψ_C corresponding to leading eigenvalues.

The projection matrix in $\bar{\mathbb{F}}$ is defined as:

$$\mathbf{W}_C = (\mathbf{I} - \Phi\Phi^T)\Psi_C. \quad (2.20)$$

4. Then project the sample $\mathbf{h}$ and class mean $\bar{\mathbf{h}}_i$ into $\mathbb{F}$ as $\mathbf{p} = \mathbf{W}_P^T \mathbf{h}$ and $\bar{\mathbf{p}}_i = \mathbf{W}_P^T \bar{\mathbf{h}}_i$, and into $\bar{\mathbb{F}}$ as $\mathbf{c} = \mathbf{W}_C^T \mathbf{h}$ and $\bar{\mathbf{c}}_i = \mathbf{W}_C^T \bar{\mathbf{h}}_i$. The classifier that minimizes the classification error decides that a sample belongs to class i if the following distance is minimized:

$$d_i = |\mathbf{p} - \bar{\mathbf{p}}_i|^2 - |\mathbf{c} - \bar{\mathbf{c}}_i|^2 / \rho. \quad (2.21)$$

Dual-space LDA makes use of the discriminant information in the full feature space. However, two subspaces have different distance metrics, i.e. in the principal subspace the Mahalanobis distance is used whereas in the null subspace the Euclidean distance is used. It is not suitable to directly combine the distance measures in these two subspaces. ρ serves to balance these two distance measures. However, Eqn.(2.21) may not be the optimal way to combine the distance measures. Bayesian Maximum Likelihood in Section 2.1.6 and ERE approach in Section 2.1.7 provide a better fusion scheme.

More importantly, the eigenspectrum in the principal subspace may not be reliably estimated. As pointed out in [12], the small eigenvalues in the principal subspace of Σ_w are also severely biased due to limited training samples. Jiang [12] partitioned the whole eigenspectrum of Σ_w into three subspaces: reliable principal subspace, unreliable principal subspace and null subspace. Each subspace is treated different and a novel eigenspectrum model is proposed. More details can be found in Section 2.1.7.

2.1.6 Bayesian Maximum Likelihood

Different from the approaches discussed earlier, in Bayesian Maximum Likelihood approach [28] the distribution of the sample-difference is modeled instead of the sample itself. Moghaddam et al. [28] proposed a probabilistic similarity measure based on the image intensity differences,

denoted by $\Delta = \mathbf{h}_1 - \mathbf{h}_2$, which characterizes the image variances of a subject. The image differences are classified as one of two classes: intra-class variations Ω_I that represents the sample differences from the same class and extra-class variations Ω_E that represents the sample differences from different classes. The similarity measure is defined as:

$$S(\mathbf{h}_1, \mathbf{h}_2) = P(\Omega_I|\Delta), \quad (2.22)$$

where $P(\Omega_I|\Delta)$ is posterior probability. Based on Bayes rule,

$$S(\mathbf{h}_1, \mathbf{h}_2) = \frac{P(\Delta|\Omega_I)P(\Omega_I)}{P(\Delta|\Omega_I)P(\Omega_I) + P(\Delta|\Omega_E)P(\Omega_E)}, \quad (2.23)$$

where $P(\Omega_I)$ and $P(\Omega_E)$ are prior probabilities, $P(\Delta|\Omega_I)$ and $P(\Delta|\Omega_E)$ are likelihoods.

Alternatively, the similarity measure can be defined in a simpler form as:

$$S' = P(\Delta|\Omega_I), \quad (2.24)$$

which leads to *Maximum likelihood* (ML) recognition. Moghaddam [28] shows that ML measure is almost as effective as MAP in most cases. It decomposes the face image space into a principal subspace $\mathbb{F}$ and a residual subspace $\bar{\mathbb{F}}$. $P(\Delta|\Omega_I)$ is then estimated as the product of two independent marginal Gaussian densities, respectively, in $\mathbb{F}$ and $\bar{\mathbb{F}}$ as,

$$P(\Delta|\Omega_I) = \frac{\exp(-\frac{1}{2} \sum_{k=1}^m y_k^2 / \lambda_k)}{(2\pi)^{m/2} \prod_{k=1}^m \lambda_k} \times \frac{\exp(-\frac{1}{2\rho} \varepsilon^2(\Delta))}{(2\pi\rho)^{(n-m)/2}}, \quad (2.25)$$

where λ_k is the k -th largest eigenvalue of the intra-personal covariance matrix, y_k is the k -th leading principal component and

$$\varepsilon^2(\Delta) = \|\Delta\|^2 - \sum_{k=1}^m y_k^2. \quad (2.26)$$

ρ is estimated as the average of eigenvalues in $\bar{\mathbb{F}}$, i.e.

$$\rho = \frac{1}{N-m} \sum_{k=m+1}^N \lambda_k. \quad (2.27)$$

The sufficient statistic of $P(\Delta|\Omega_I)$ defined in Eqn. (2.25) is:

$$d(\Delta) = \sum_{k=1}^m \frac{y_k^2}{\lambda_k} + \frac{\varepsilon^2(\Delta)}{\rho}. \quad (2.28)$$

As the eigenvalues in the residual subspace $\bar{\mathbb{F}}$ are noisy and unreliable, ρ is problematic and hence affects the reliable classification. Furthermore, ρ estimated in Eqn. (2.27) leads to a very small constant compared to the eigenvalues in $\mathbb{F}$. Thus, the features in $\bar{\mathbb{F}}$ is much more heavily weighted than the features in $\mathbb{F}$.

In [11], Jiang et al. used ρ_{up} to replace ρ in Eqn. (2.28), which is defined as:

$$\rho_{up} = \max\{\lambda_k | k > m\}. \quad (2.29)$$

Here, ρ_{up} is estimated using the last reliable eigenvalues in $\mathbb{F}$. Thus, it is much more reliable than ρ estimated using unreliable eigenvalues in $\bar{\mathbb{F}}$. Besides, ρ_{up} is larger than ρ . Thus, a smaller weight is used for the features in the unreliable subspace $\bar{\mathbb{F}}$. ρ_{up} partially alleviates the over-fitting problem caused by the small number of training samples w.r.t. the high feature dimensionality. However, it still has two drawbacks: 1) It does not differentiate noisy principal subspace from the null subspace. These two subspaces deserves different treatments. 2) ρ_{up} may be biased upwards, as it uses the largest eigenvalues in $\bar{\mathbb{F}}$ to weight all the features in $\bar{\mathbb{F}}$. Consequently, a significantly smaller weight is used for these features. As a result, the features in $\bar{\mathbb{F}}$ may be easily overwhelmed by the features in $\mathbb{F}$. ERE approach [12] discussed in the next section better solves these problems.

2.1.7 ERE Approach

In the ERE approach [12], the eigenspectrum of Σ_w is divided into three subspaces: reliable principal subspace $\mathbb{F} = \{\phi_k^w\}_{k=1}^m$, noisy principal subspace $\mathbb{N} = \{\phi_k^w\}_{k=m+1}^r$ and null subspace $\emptyset = \{\phi_k^w\}_{k=r+1}^d$, where r is the rank of Σ_w , and ϕ_{m+1}^w is the starting dimension of unreliable principal subspace. $\mathbb{F}$, $\mathbb{N}$ and $\emptyset$ are spanned by the dimensions corresponding to the large, small

and zero eigenvalues, respectively. The large eigenvalues are reliably estimated whereas the small eigenvalues in $\mathbb{N}$ and zero eigenvalues in $\emptyset$ are problematic. As shown in Eqn. (2.6), the inverse of the eigenvalues are used to weight the features. The small eigenvalues deviate far from their true values [12], which causes a severe overfitting problem on novel testing data. In addition, the zero eigenvalues in $\emptyset$ causes that Eqn. (2.6) cannot be evaluated. To solve these problems, an eigenspectrum model was proposed in [12] to replace the eigenspectrum of Σ_w . Each subspace is treated differently. The eigenvalues in $\mathbb{F}$ are reliable and remain unchanged. The eigenvalues in $\mathbb{N}$ are estimated using the reliable eigenvalues in $\mathbb{F}$. The eigenvalues in $\emptyset$ are replaced by the smallest estimated eigenvalue in $\mathbb{N}$. Then, the classifier in Eqn. (2.6) is modified as:

$$g_i(\mathbf{h}) = \sum_{j=1}^d \frac{(y_j - y_{i,j})^2}{\tilde{\lambda}_j^w} + b_i, \quad (2.30)$$

where $\tilde{\lambda}_j^w$ is the eigenvalue of the new eigenspectrum model.

2.1.8 Summary and Discussion

Subspace approaches originate from the Bayesian classification. Under the Gaussian assumption, the optimal decision rule is simplified to evaluating the Mahalanobis distance defined in Eqn. (2.5) or Eqn. (2.6). Due to limited training samples and the high feature dimensionality, the eigenspectrum of Σ_w is poorly estimated, especially the small eigenvalues, which adversely affects the reliable classification. Various subspace approaches have been proposed to address this problem. Eigenface [6] removes the dimensions corresponding to the small unreliable eigenvalues, and hence improves the generalization performance on novel testing data. PCA is not designed for optimal classification, and hence asymmetric PCA [24] was proposed to assign a larger weight to the unreliable covariance matrix so that its unreliable dimensions can be better removed. Fisherface [7] improves the generalization performance by removing the unreliable dimensions using PCA. Then, it extracts the discriminant information in a small

PCA+LDA subspace. The null subspace of Σ_w also contains important discriminant information. Thus, in [8] the null subspace of Σ_w is utilized for classification. In fact, both principal and null subspaces can be used for classification. In dual-space LDA [9], the discriminant information in both subspaces is utilized. Different from these approaches, in Bayesian Maximum Likelihood [28] and Enhanced Maximum Likelihood [11], the sample difference instead of the sample itself is modeled. The image space is divided into a principal subspace and a residual subspace spanned by the dimensions corresponding to large and small eigenvalues, respectively. Then, the discriminant information in both subspaces is extracted. In fact, as shown in ERE approach [12], the image space should be divided into three subspaces: reliable principal subspace $\mathbb{F}$, noisy subspace $\mathbb{N}$ and null subspace $\emptyset$ corresponding to large, small non-zero and zero eigenvalues, respectively. Each subspace is treated differently. ERE approach regularizes the eigenspectrum of Σ_w to alleviate the overfitting problem caused by the unreliable estimation of the eigenspectrum.

Subspace approaches utilizing up to the second-order statistic are optimal when the underlying distribution is Gaussian. However, we find that LBP features deviate far from Gaussian distribution, and hence we propose to transform them to a new feature space so that the resulting feature fits better to Gaussian distribution. This will be illustrated in Section 7.

2.2 Local Binary Pattern

LBP and its variants have been widely used in many recognition applications, e.g. texture classification [16, 26, 29, 30], dynamic texture recognition [19, 31, 32], facial analysis [1, 17, 27, 33], scene classification [3, 34, 35] and others [36–38]. LBP is popular because of the following desirable properties: insensitive to monotonic illumination variations, robust to alignment error, simple and fast to extract.

Local binary pattern encodes the relative pixel difference of the neighboring pixels to the

central pixels, e.g. $LBP_{P,R}$ denotes the LBP code of P neighbors at the distance of R . Denote the gray level of the center pixel by i_c , and the gray level of the circular neighbors by i_p , where $p = 0, 1, \dots, P - 1$. The LBP code for pixel (x, y) is obtained as:

$$LBP_{P,R}(x, y) = \sum_{p=0}^{P-1} s(i_p - i_c) 2^p, \quad (2.31)$$

where $s(z)$ is the threshold function

$$s(z) = \begin{cases} 1 & \text{if } z \geq 0, \\ 0 & \text{if } z < 0. \end{cases} \quad (2.32)$$

The histogram of these codes is then used for further analysis.

Despite its wide applications, LBP features still have some limitations. We focus on two main problems. 1) LBP is sensitive to image noise. A small image variation will alter the LBP bit from 0 to 1 or vice versa, and hence alter the code significantly. Many approaches have been proposed to address this issue. We review them in Section 2.2.1. 2) To preserve the locality, an image is often divided into patches and one histogram is extracted from each patch. To capture the image details at different scales, multi-scale LBP is often utilized. All the histograms are then concatenated to form the final feature vector. This often results in a high-dimensional feature vector. In addition, recent researches [2, 16, 18] show that LBP features extracted in a large neighborhood using more bits will improve the recognition performance, at the cost of high dimensionality. In Section 2.2.2, we review the approaches that alleviate the high-dimensionality problem of LBP.

In addition, the LBP features are often not reliably estimated due to insufficient pixels, especially for those infrequent patterns. Subspace approaches are often applied to improve the reliability of LBP features. In Section 2.3 we review the subspace approaches applied on LBP features.

2.2.1 LBP Variants Robust to Image Noise

2.2.1.1 Uniform LBP

It is shown in [16, 17] that most LBP codes in a natural image are uniform codes. A LBP code is defined as uniform pattern if it has at most two circularly bitwise transitions from 0 to 1 or vice versa, and non-uniform patterns if otherwise. For example, “11000000” is a uniform code whereas “11001100” is a non-uniform code. For $LBP_{8,2}$, there are 58 uniform codes and 198 non-uniform codes. Uniform patterns are statistically significant, and their occurrence probabilities can be reliably estimated. In contrast, non-uniform patterns are statistically insignificant, and hence noise-prone and unreliable. In uniform LBP mapping [16], one separate histogram bin is used for each uniform pattern and all non-uniform patterns are accumulated in a single bin. By grouping the non-uniform patterns into one label, the noise in non-uniform patterns is suppressed. The number of patterns is reduced significantly at the same time.

In [3, 29, 39–41], information in non-uniform patterns is extracted and used for classification. Liao et al. proposed dominant LBP (DLBP) that considers the most frequently occurred patterns in a texture image [29]. This is based on the observation that some non-uniform codes may have higher occurrence frequency than uniform codes. Instead of based on image microstructures [16], DLBP chooses the LBP codes based on occurrence frequency directly. Zhou et al. [39] and Fathi et al. [40] proposed to extract information from non-uniform patterns based on pattern uniformity measure and the number of ones in the LBP codes. Principal component analysis [3] and random subspace approach [41] were utilized to extract information from the whole LBP histogram. These approaches can extract discriminant information from the non-uniform codes. However, as the non-uniform codes may contain noise, these approaches tend to be sensitive to noise.

2.2.1.2 Fuzzy LBP

“Soft histogram” is another approach to improve the robustness to noise. Instead of hard-coding the pixel difference as shown in Eqn. (2.32), a probability measure is utilized to represent its likelihood as 0 or 1. In [42, 43], a fuzzy LBP (FLBP) utilizing piecewise linear fuzzy membership function was proposed, which replaces $s(z)$ with two fuzzy membership functions:

$$f^1(z, d) = \begin{cases} 0 & \text{if } z < -d, \\ 0.5 + 0.5z/d & \text{if } -d \leq z \leq d, \\ 1 & \text{if } z > d. \end{cases} \quad (2.33)$$

$$f^0(z, d) = 1 - f^1(z, d). \quad (2.34)$$

The parameter d controls the amount of fuzzification. $f^1(z, d)$ and $f^0(z, d)$ represent the probability that the pixel difference z is encoded as 1 and 0, respectively. When constructing the LBP histogram, we calculate the probabilities of all patterns as:

$$P_j = \prod_{i=0}^{P-1} c_i f_i^1(z, d) + (1 - c_i) f_i^0(z, d), \quad (2.35)$$

where $j = \sum_{i=0}^{P-1} c_i \times 2^i$ and c_i is i -th bit of the code. For example, the probability of “11001100” is calculated as: $P_{11001100} = f_7^1 f_6^1 f_5^0 f_4^0 f_3^1 f_2^1 f_1^0 f_0^0$, where f_i^b is the probability of bit i to be encoded as $b \in \{0, 1\}$. The complete histogram is derived by summing the contributions of all pixels in an image patch.

Another fuzzy LBP utilizes Gaussian-like membership function [44], which is defined as:

$$f^1(x, \sigma) = \begin{cases} 1 - 0.5e^{-x^2/\sigma^2} & \text{if } x \geq 0, \\ 0.5e^{-x^2/\sigma^2} & \text{if } x < 0, \end{cases} \quad (2.36)$$

$$f^0(x, \sigma) = 1 - f^1(x, \sigma) \quad (2.37)$$

where $x = (i_p - i_c)/i_c$ and σ controls the amount of fuzzification. A comprehensive comparison between LBP and fuzzy LBP in classifying and segmenting textures can be found in [45].

When the original LBP operator is used, one pixel only contributes to one histogram bin. In fuzzy LBP, one pixel typically contributes to more than one bin. The contributions of a pixel to all bins always sum to 1. By using fuzzy LBP, a small change in the input image only causes a small change in the histogram. This alleviates the noise-sensitivity problem to a certain extent. However, the probability measures defined in Eqn. (2.33) and Eqn. (2.36) are closely related to the magnitude of the pixel difference. Thus, it is still sensitive to noise.

2.2.1.3 LTP and Its variants

The LBP code is sensitive to noise, especially in a near-constant image region. In local ternary pattern (LTP) [1], the pixel difference z between a pixel and its neighbors is encoded by three values (1, 0 or -1) according to a threshold t , i.e.

$$s_T(z) = \begin{cases} 1 & \text{if } z > t, \\ 0 & \text{if } -t \leq z \leq t, \\ -1 & \text{if } z < -t. \end{cases} \quad (2.38)$$

State “0” is used to handle the small image variations, i.e. if the difference is within the threshold t , it is encoded as state “0”. Those small variations are treated separately. Thus, LTP is less sensitive to noise. However, its feature dimensionality is very high, e.g. $LTP_{8,2}$ exhibits a histogram of $3^8 = 6561$ bins. To reduce the dimensionality, the ternary pattern is split into a positive LBP code and a negative LBP code in [1].

Subsequently, many LTP variants were proposed in the literature. Nanni et al. proposed a

quinary code of five values according to two thresholds [46],

$$s_q(z) = \begin{cases} 2 & \text{if } z \geq t_2, \\ 1 & \text{if } t_1 \leq z < t_2, \\ 0 & \text{if } -t_1 \leq z < t_1, \\ -1 & \text{if } -t_2 \leq z < -t_1, \\ -2 & \text{if } z < -t_2, \end{cases} \quad (2.39)$$

where t_1, t_2 are two thresholds and $t_1 < t_2$. Then, the quinary code is split into four binary codes similarly as for LTP. As LTP is not invariant under scaling of intensity values, Liao et al. proposed scale-invariant local ternary pattern (SILTP) to deal with the gray scale intensity changes in a complex background [37]. To reduce the high dimensionality of LTP, center-symmetric LTP (CSLTP) was proposed in [20]. Instead of using the pixel differences between the neighboring pixels and the central pixel, the pixel differences between diagonal neighbors are calculated. In local adaptive ternary patterns (LATP) [47] and extended LTP (ELTP) [48], instead of using a constant threshold, the threshold is calculated for each window using local statistics, which makes them less sensitive to illumination variations. In local triplet pattern [49], the equality is modeled as a separate state, and a tri-state pattern is formulated. It can be viewed as a special case of LTP [1].

To improve the robustness to noise, other thresholding and encoding techniques have been proposed. Instead of compared with the center pixel, Hafiane et al. proposed median binary pattern (MBP) by thresholding the local pixel values against the median value within the local neighborhood [50]. In improved LBP [51], the neighboring pixels are threshold against the mean intensity of the local neighborhood. Trefny and Matas [52] proposed two new encoding schemes: binary value transition coded LBP (tLBP) utilizing the pixel differences of neighbors in clockwise direction and direction coded LBP (dLBP) encoding the second-order pixel differences. In centralized binary pattern (CBP) [53], the average intensity within the neighborhood

is compared with the center pixel and added as a LBP bit. To reduce the sensitivity to noise, a Bayesian LBP (BLBP) was developed by He et al. [54]. It is formulated in a filtering, labeling and statistic (FLS) framework, in which the labeling procedure is modeled as an optimization process.

2.2.1.4 Preprocessing

Image pre-processing techniques can also improve the robustness of LBP features to image noise. Gabor filtering is often used before extracting LBP features, as LBP features capture image micro-structures and Gabor filters encode image information in a large scale. Zhang et al. [55] proposed to extract LBP features from 40 Gabor-filtered images, known as local gabor binary patterns (LGBP). It has been used as a benchmark method in many recent face recognition studies. The drawback is its high dimensionality. Tan and Triggs [1] proposed an image pre-processing framework that consists of gamma correction, difference of Gaussian (DoG) filtering, masking (optional) and equalization of variations. Used together with LTP, it has demonstrated a superior performance in face recognition.

To handle the small image variations in a near-constant region, Yao and Chen [56] proposed local edge patterns (LEP) for color texture retrieval, in which the Sobel edge detection is used to suppress the small image variations in the near-constant region, and LBP-like feature is extracted on the filtered image. Similar ideas have been used in [57] for shape localization and in [58] for facial image representation. Li et al. [59] proposed to extract LBP features from multi-scale heat kernels, which capture the intrinsic topological structural information of face appearance. In [60], LBP features are extracted from curvelet-transformed images for medical image analysis.

2.2.1.5 Summary and Discussion

Many approaches have been proposed to address the noise-sensitivity problem. However, those approaches treat the corrupted patterns as they are, and they lack a mechanism to recover the underlying patterns. Therefore, in Chapter 4, we propose a noise-resistant LBP with an embedded error-correction mechanism to better handle image noise.

2.2.2 Reducing Feature Dimensionality of LBP

The LBP features are often of high dimensionality because: 1) To preserve the locality, an image is often divided into patches and one histogram is extracted in each patch [1, 17, 27]. These patch-based LBP histograms are concatenated to form a feature vector of high dimensionality. 2) Multi-scale LBP [61–63] is often used to extract image characteristics at different scales. This also significantly increases the dimensionality. 3) LBP features extracted in a larger neighborhood using more bits can improve the performance, at the price of high dimensionality [2, 16, 18]. 4) LBP feature extracted on filtered images may also increase the dimensionality, e.g. in [55], one image results in 40 gabor-filtered images, and hence the resulting dimensionality is 40 times of that of original LBP features.

In a recent study [63], high-dimensional LBP feature extracted on a patch-wise basis at multiple scales has achieved the state-of-the-art performance on face recognition. However, its feature dimensionality needs to be reduced to a reasonable range before further processing.

In this section, we review the approaches that handle the high-dimensionality problem, which can be classified into three categories: 1) Direct feature selection/extraction from LBP-histogram features [2, 16, 18, 29, 39, 64–72], discussed in Section 2.2.2.1. In these approaches, feature selection/extraction techniques are applied on the high-dimensional feature vector to select/extract the discriminant information. The drawback is that in some cases, the LBP structure may be too large to enumerate all the histogram bins, as the number of histogram bins

increases exponentially with the number of bits. 2) Handcrafted LBP structures of a smaller size [2, 18, 19, 36, 46, 72–77], discussed in Section 2.2.2.2. In these approaches, the dimensionality is reduced by breaking the large LBP structure into small ones, or simply replacing it by a small one. However, the handcrafted structures may not be optimal. 3) Instead of using handcrafted structures, some approaches [78, 79] aim to find the optimal LBP structure in a data-driven way, discussed in Section 2.2.2.3.

Subspace approaches can also reduce the feature dimensionality, which are discussed separately in Section 2.3.

2.2.2.1 Direct-Bin Selection/Extraction

In uniform LBP [16], all non-uniform patterns are grouped into one bin in order to suppress the noise in LBP histogram. At the same time, the dimensionality is significantly reduced from 2^P to $P(P - 1) + 3$. In Dominant LBP [29], the LBP codes are sorted according to their occurrence frequencies. It removes the noise in LBP histogram and reduces the dimensionality by only keeping the frequent patterns. Mäenpää et al. [66] showed that the discriminative power concentrates in a small subset of patterns. They utilized beam search to find the discriminative pattern sets in an iterative way. In local quantized pattern (LQP) [2, 18], k-means clustering algorithm is used to cluster the histogram bins into codewords.

Lahdenoja et al. [64] defined a discrimination concept of symmetry for uniform patterns to reduce the feature dimensionality. The level of symmetry S of a LBP code is defined as

$$S = \min\left\{\sum_{i=0}^{P-1} c_i, \sum_{i=0}^{P-1} \bar{c}_i\right\}, \quad (2.40)$$

where c_i is i -th bit of the LBP code. The most symmetric pattern is always the symmetric edge detector consisting of equal number of ones and zeros. The patterns with a higher level of symmetry are shown to have more discriminative power.

In [39], a similarity measure is defined for each non-uniform pattern, which is the minimum

Hamming distance to map it to a uniform code. Then, non-uniform patterns are classified and accumulated into different histogram bins. Similarly, Yang and Wang [65] proposed to replace non-uniform patterns by uniform patterns with minimum Hamming distance.

Smith and Windeatt [67] used the fast correlation-based filtering (FCBF) algorithm to select the most discriminative patterns, in which relevant features as well as redundancy among relevant features are derived without pairwise correlation analysis. The information-theoretic concept of symmetric uncertainty is used as the correlation measure. Guo et al. [68] proposed a Fisher separation criterion to learn the most reliable and robust patterns using intra-class and inter-class distances. The most reliable patterns for each class are determined, and then merged to form the global dominant set.

Boosting is widely adopted for LBP feature selection in various tasks, e.g. 3D face recognition [69], face detection [70] and gender classification [80]. Boosting can be used to select the system parameters, e.g. Zhang et al. [70] used AdaBoost to determine the optimal weights of image patches for face recognition. It can be also used to select histogram bins. Shan and Gritti [71] used AdaBoost to learn most discriminative LBP-histogram bins for facial expression recognition. In [72], a shrink boost is used to select multi-LBP-histogram features in object detection.

2.2.2.2 Handcrafted LBP Structures

To reduce the dimensionality, the large LBP structure is often split into small handcrafted ones that utilizes only a subset of neighbors. In elliptical binary pattern [73], instead of using circular neighbors, elliptical neighbors are used to construct the LBP histogram. Nanni et al. [46] tested different neighborhood topologies (circle, ellipse, parabola, hyperbola and Archimedean spiral) and encoding schemes (binary, ternary and quinary) for medical image analysis. Quinary encoding in an elliptic neighborhood performs best. Petpon and Srisuk [74] proposed to use lines in vertical and horizontal directions to form the LBP structure. In Center-Symmetric LBP

(CS-LBP) [36], instead of thresholding against the center pixel, the pixel difference between diagonal neighbors are utilized. In [72], three LBP quantization schemes were proposed by utilizing the pixel differences among the neighbors in a neighborhood of 3×3 pixels. In local quantized pattern [2, 18], a subset of pixels in a large neighborhood are compared with the center pixel and used to construct the LBP histogram. Many sampling schemes are considered, e.g. vertical line, horizontal line, cross and disc shape.

In multiscale block local binary pattern (MB-LBP) [75], the neighborhood is split into small ones, e.g. 9×9 local neighborhood is divided into 3×3 non-overlapped sub-neighborhoods. Then, the mean value of each sub-neighborhood is compared with the one at the center. In [76], the sub-neighborhood similarity is used to construct the LBP histogram rather than using the differences of mean intensity values.

Mu et al. [77] proposed two variants of LBP: Semantic-LBP (S-LBP) and Fourier-LBP (F-LBP). In S-LBP, a joint histogram of arch (connected ones) length and arch angle is built. In F-LBP, the color distance between a pixel and its neighbors is computed and transformed into frequency domain. The low-frequency components are used to capture image micro-structures.

The spatial-temporal LBP often produces a feature vector of very high dimensionality, e.g. volume-LBP [19] considers P circular neighbors in 3 successive frames, which has a histogram of 2^{3P+2} bins. In LBP-TOP [19], this large spatial-temporal LBP structure is broken into small handcrafted ones, i.e. the LBP features are extracted from three orthogonal planes: $x - y$, $x - t$ and $y - t$ planes. Then, these histograms are concatenated to form the final feature vector.

The handcrafted structures have some limitations: a) Non-optimal as they are often selected heuristically. b) Non-scalable as it is difficult to define a consistent rule to build handcrafted structures of different sizes. c) Non-universal as the optimal LBP structure should be task-dependent in order to capture the intrinsic image characteristics of different applications. In the next subsection, we will review the approaches of learning LBP structures.

2.2.2.3 LBP Structure Learning

In [81], a deep neural network learned by fitting it to a huge amount of training data is used to extract features. This data-driven feature outperforms engineering features. Similarly, a properly learned LBP feature can outperform traditional LBP features. Given a large pool of LBP bits, the objective is to find an optimal subset of LBP bits to form the LBP structure.

In discriminative LBP [78], the LBP structure is optimized in a data-driven way. As it is difficult to find the globally optimal solution, a heuristic hill-climbing technique was utilized. Initially, a subset is randomly chosen as a starting point. In each iteration, one LBP bit is randomly chosen and replaced by another random bit. A Fisher-like criterion J is used to determine whether it should be replaced,

$$J = \frac{(\mu_w - \mu_b)^2}{\sigma_w^2 + \sigma_b^2}, \quad (2.41)$$

where μ_w and μ_b are the mean within-subject and between-subject distances and σ_w and σ_b are the variances of the within-subject and between-subject distances, respectively. Such a hill-climbing technique cannot guarantee the optimal LBP structure.

Lei et al. proposed a discriminant face descriptor (DFD) by learning discriminant image filters and neighborhood sampling strategy in a data-driven way [79]. They formulate these two tasks as two generalized eigen-decomposition problems, and derive the optimal solution in an iterative way, i.e. when solving one generalized eigen-decomposition problem, the parameters in the other problem are treated as constant. DFD demonstrates a superior performance on face recognition.

2.2.2.4 Summary and Discussion

The drawback of direct bin selection/extract approaches is that it needs to enumerate all histogram bins before further processing, which may be infeasible for a large LBP structure. Thus,

the large LBP structure is often split into small handcrafted ones. In many situations, the handcrafted structures may not yield a desirable result as it is often selected heuristically. More importantly, the optimal structure should be task-dependent because the intrinsic image characteristics of different applications are different. Therefore, in [78, 79], two LBP-structure-learning approaches were proposed. However, Discriminative LBP [78] utilizing a heuristic hill-climbing technique cannot guarantee structure optimality, and DFD [79] can only well handle face image variations but not large image variations such as scene images. In Chapter 5, we propose a new method to learn the optimal LBP structure towards the objective of reducing the feature dimensionality and extracting discriminant information.

2.3 Subspace Approaches on LBP Features

Subspace approaches have been utilized to derive a compact and discriminative feature vector, which project the LBP features from the original high-dimensional feature space into a low-dimensional subspace. In this section, we briefly review some subspace approaches applied on LBP features.

2.3.1 Principal Component Analysis on LBP Features

Principal component analysis (PCA) has been applied on LBP features to derive a compact feature representation. In CENTRIST feature [3], PCA is applied on the LBP histogram of each patch to reduce the dimensionality. Then, the features of all patches are concatenated to form the final feature vector and classified by a support vector machine (SVM) with a RBF kernel. CENTRIST feature achieves a good performance on scene recognition. CENTRIST was improved in [82] by mining discriminative co-occurrence patterns. In [35], CENTRIST features are extracted in multiple color channels, and combined to form the final feature vector. Multi-channel CENTRIST [35] greatly improves the performance on scene recognition.

In [83], a random projection tree was utilized to encode LBP-like features, and PCA was applied to reduce the dimensionality. To derive a compact feature representation, PCA can also be applied on the concatenated LBP histograms of all patches [22, 63, 84]. In [85], PCA is used to reveal the hidden structures of multi-level HOG-LBP feature by projecting the features from a high-dimensional feature space to a low-dimensional subspace. This HOG-LBP detector makes the head-shoulder detection more robust. Multi-scale, multi-patch LBP feature may have a very high dimensionality, e.g. the dimensionality of LBP features in [63] is 100k. PCA was thus applied to derive a compact feature representation. This high-dimensional LBP feature has demonstrated a superior performance in face recognition.

2.3.2 Linear Discriminative Analysis on LBP Feature

Linear discriminant analysis (LDA) has been used to derive discriminative LBP features. In [21], PCA was applied on the multi-radius LBP features to remove the unreliable dimensions, and then LDA was used to extract the discriminative features, in a similar way as shown in Section 2.1.3. Chan et al. [86] used LDA to extract the discriminative features from multi-scale LBP features.

In [87], a large pool of multi-scale LBP features was generated by shifting and scaling the sub-window over the image. For each LBP histogram, PCA+LDA was applied to extract the discriminative features and a nearest-neighbor classifier is built as the weak classifier. Then, Adaboost algorithm was used to select the discriminative features and boost the weak classifiers into a strong classifier.

To deal with the small-sample-size problem of LDA, Shan et al. [88] proposed an ensemble of piecewise Fisher Discriminant Analysis (EPFDA) classifiers to reduce the dimensionality of local Gabor binary pattern (LGBP). As the dimensionality of LGBP is 40 times that of the LBP, it may cause serious curse of dimensionality. A direct application of PCA+LDA will result in loss of too much information. Thus, the high-dimensional histogram of local Gabor binary

pattern (LGBP) features are divided into segments and one set of discriminative features are learned from each segment. Finally, a nearest-neighbor classifier with cosine similarity is used for classification.

In [1], PCA was applied on the feature vector built from Gabor wavelets and LBP features to reduce the dimensionality, and then the Kernel Discriminative Common Vectors (KDCV) was used to extract the discriminative features for face recognition. As shown in Section 2.1.5, both principal and null subspaces of Σ_w contain useful discriminant information. Thus, in [89], Dual-Space LDA was used to extract the discriminative LBP features from both subspaces.

2.3.3 Kernel PCA and Laplacian PCA on LBP Features

In [90], Zhao et al. compared different dimensionality-reduction techniques on LBP features, e.g. PCA, kernel PCA and Laplacian PCA. Laplacian PCA demonstrates a superior performance to PCA and kernel PCA.

In kernel-PCA, N samples $\{\mathbf{h}_i\}, i = 1, 2, \dots, N$, are non-linearly mapped into the high-dimensional space $\mathbb{F}$ as $\Phi(\mathbf{h}_i)$, i.e.

$$\Phi : \mathbb{R}^d \rightarrow \mathbb{F}. \quad (2.42)$$

Assume $\Phi(\mathbf{h}_i)$ is centered, i.e. $\sum_{i=1}^N \Phi(\mathbf{h}_i) = 0$. The kernel matrix $\mathbf{K}$ is defined by

$$\mathbf{K}_{i,j} := k(\Phi(\mathbf{h}_i), \Phi(\mathbf{h}_j)) = \Phi(\mathbf{h}_i)^T \Phi(\mathbf{h}_j). \quad (2.43)$$

After solving the eigen-decomposition problem

$$N\lambda\mathbf{K} = \mathbf{K}\alpha, \quad (2.44)$$

eigenvalues λ and eigenvectors α of $\mathbf{K}$ are derived. The projection matrix $\mathbf{V}$ in feature space $\mathbb{F}$

is derived as:

$$\mathbf{V} = \sum_{i=1}^N \alpha_i \Phi(\mathbf{h}_i). \quad (2.45)$$

Denote $\mathbf{V}^k$ as k -th dimension of $\mathbf{V}$, α^k as k -th dimension of α and λ_k as k -th eigenvalue. To satisfy $k(\mathbf{V}^k, \mathbf{V}^k) = 1$, $|\alpha^k| = 1/\sqrt{N\lambda_k}$. For a new test sample $\mathbf{x}$, it is projected into k -th dimension of $\mathbb{F}$ as:

$$\mathbf{V}^{kT} \mathbf{x} = \sum_{i=1}^N \alpha_i^k \Phi(\mathbf{h}_i)^T \Phi(\mathbf{x}) = \sum_{i=1}^N \alpha_i^k k(\mathbf{h}_i, \mathbf{x}) \quad (2.46)$$

The samples that are not linearly separable in $\mathbb{R}^d$ may now be linearly separable in $\mathbb{F}$.

The criterion of Laplacian PCA [90] is to maximize the local scatter of data instead of the global one pursued by PCA. It aims at maximizing the following criterion:

$$\max_{\mathbf{U}} \text{tr}(\mathbf{U}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{U}), \text{ s.t. } \mathbf{U}^T \mathbf{U} = \mathbf{I}, \quad (2.47)$$

where $\mathbf{X}$ the centered training data and $\mathbf{L} = \sum_{i=1}^N \mathbf{S}_i \mathbf{L}_i \mathbf{S}_i^T$ is global Laplacian. Here, $\mathbf{L}_i = \mathbf{W}_i - \frac{\mathbf{W}_i \mathbf{e} \mathbf{e}^T \mathbf{W}_i}{\mathbf{e}^T \mathbf{W}_i \mathbf{e}}$ is the local Laplacian, where $\mathbf{W}_i$ is the weighting matrix for the local scatter, $\mathbf{S}_i$ is an indication matrix for the membership of the local scatter and $\mathbf{e}$ is all-one column vector.

Laplacian PCA extends the traditional PCA by optimizing the locally weighted scatters instead of the global non-weighted scatter in PCA. It address two issues of PCA: the sensitivity to noise and the incompetence to non-Euclidean samples. Laplacian PCA is closely related to LaplacianFaces [91]. Readers may refer to [91] for more details.

2.3.4 Dimensionality Reduction by Partial Least Squares

In [92], HOG features, local binary patterns and local ternary patterns were concatenated to form a high-dimensional feature vector of 20448 dimensions. To reduce the dimensionality, Partial Least Squares (PLS) was used to select the discriminative features, which produces a fast and efficient visual object detector.

PLS is an iterative algorithm to build low-dimensional orthogonal projections. Formally, it finds a linear regressor of the form $\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{E}$ by a low-dimensional projection $\mathbf{T} = \mathbf{X}\mathbf{W}$ followed by regression $\mathbf{Y} = \mathbf{T}\mathbf{Q} + \mathbf{E}$, so that $\mathbf{B} = \mathbf{W}\mathbf{Q}$. $\mathbf{X}$ and $\mathbf{Y}$ are input and target matrices with each row representing one example, $\mathbf{W}$ is the orthogonal projection matrix, $\mathbf{Q}$ is the corresponding regressor, and $\mathbf{E}$ is the residual error matrix. The algorithm iteratively evaluates the best new column to be added to current $\mathbf{W}$ and $\mathbf{Q}$ in order to minimize the regression error.

2.3.5 Locality Preserving Projections on LBP Feature

In [93], Locality Preserving Projections (LPP) was explored to learn expression manifolds from two feature spaces: raw image data and local binary patterns, which embeds the high-dimensional appearance feature into a low-dimensional manifold. Given a set $\mathbf{x}_i \in \mathbb{R}^d, i = 1, 2, \dots, N$, the objective is to find a transformation matrix $\mathbf{w}$ that maps these N points to $\mathbf{y}_i \in \mathbb{R}^M (M \ll d), i = 1, 2, \dots, N$, i.e. $\mathbf{y}_i = \mathbf{w}^T \mathbf{x}_i$. The optimal projections preserving locality can be obtained by solving the following minimization problem:

$$\min_{\mathbf{w}} \sum_{i,j} (\mathbf{w}\mathbf{h}_i - \mathbf{w}\mathbf{h}_j)^2 S_{i,j}, \quad (2.48)$$

where $S_{i,j}$ evaluates the local structure of data space, i.e.

$$S_{i,j} = \begin{cases} e^{-\frac{|\mathbf{h}_i - \mathbf{h}_j|^2}{t}} & \text{if } \mathbf{h}_i \text{ and } \mathbf{h}_j \text{ are close,} \\ 0 & \text{otherwise.} \end{cases} \quad (2.49)$$

The objective function in Eqn. (2.48) can be re-written as:

$$\min_{\mathbf{w}} \mathbf{w}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{w}, \text{ s.t. } \mathbf{w}^T \mathbf{X} \mathbf{D} \mathbf{X}^T \mathbf{w} = 1, \quad (2.50)$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$, $\mathbf{D}$ is a diagonal matrix whose entries are column sums of $\mathbf{S}$ and $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is Laplacian matrix. The solution to the above optimization problem can be derived

by solving the following generalized eigen-decomposition problem:

$$\mathbf{X}\mathbf{L}\mathbf{X}^T \mathbf{w} = \lambda \mathbf{X}\mathbf{D}\mathbf{X}^T \mathbf{w}. \quad (2.51)$$

Although LPP is a linear technique, it uncovers the intrinsic nonlinear manifold structures by preserving local structures.

2.4 Summary

Subspace approaches have been widely applied on holistic features to derive a compact and discriminative feature representation. They originate from Bayesian classification and are optimal only when the underlying distribution is Gaussian. Under Gaussian assumption, the Bayes' classifier is simplified to a minimum-Mahalanobis-distance classifier. Due to limited training samples and high feature dimensionality, this Mahalanobis distance cannot be reliably calculated in the high-dimensional feature space. Thus, various subspace approaches have been proposed to address this problem [6–12]. Among them, ERE approach [12] performs best. It decomposes the feature space into three subspaces and each subspace is treated differently. It improves the performance on the novel testing data by regularizing the eigenspectrum of Σ_w . Large-matrix multiplication is often required for subspace approaches. Thus, it may not be suitable for detection task where millions of sliding windows need to be scanned. In Chapter 3, we utilize attentional cascaded structure to build a fast and accurate subspace face/eye detector, and then develop a fully automated face verification system on mobile devices.

Local binary pattern is one of most popular features in visual recognition because of its robustness to alignment error and illumination variations. However, LBP features still have some limitations.

Firstly, LBP is sensitive to image noise. A small image variation may alter the LBP code significantly. Various approaches have been proposed to address this issue. In Uniform-LBP [16],

the image noise is suppressed by grouping noisy non-uniform patterns into one bin. In fuzzy LBP [42–44], a fuzzy membership function is defined for pixel difference. The image noise will only alter slightly the histogram of fuzzy LBP. In local ternary pattern [1], the small pixel difference is encoded as a separate state and hence LTP is more robust to image noise. Pre-processing techniques such as Gabor filtering [55], DoG filtering [1] and local edge patterns [56] may also improve the robustness to the image noise. However, all these approaches do not attempt to recover the corrupted local binary patterns. In Chapter 4, we propose a noise-resistant local binary pattern to address this issue.

Secondly, multi-scale multi-patch LBP may have a very high dimensionality. The dimensionality needs to be reduced to a reasonable range before further processing. In [2, 16, 18, 29, 39, 64–72], feature selection/extraction methods were utilized to extract discriminative LBP features. However, in some cases it may not be feasible to enumerate a very large LBP histogram. Thus, the large LBP structure was broken into small handcrafted ones [2, 18, 19, 36, 46, 72–77]. However, handcrafted structures are not optimal as they are often selected heuristically. In structure-learning approaches [78, 79], the optimal LBP structure is derived in a data-driven way. However, the heuristic hill-climbing technique [78] may not guarantee optimality and DFD [79] may not handle large image variations. In Chapter 5, we propose a new method to directly optimize the LBP structure.

LBP features may not be reliably estimated due to insufficient pixels and its unbalance nature. To address this problem, subspace approaches are often applied, e.g. PCA [3, 22, 63, 83–85], LDA [1, 21, 86, 88] and Laplacian PCA [90]. In Chapter 6, we propose two learning-based LBP features to better remove the unreliable dimensions in LBP features. We further find that LBP features deviate far from Gaussian distribution, which limits the performance gain brought by subspace approaches as they are optimal under Gaussian assumption. We thus propose a novel transformation so that the resulting feature fits better to Gaussian. This will be discussed in detail in Chapter 7.

Chapter 3

A Complete and Fully Automated Face Verification System on Mobile Devices

In this chapter, we present a natural and non-intrusive way to secure mobile devices, i.e. a complete and fully automated face verification system ¹. It consists of three sub-systems: face detection, alignment and verification. The proposed subspace face/eye detector locates the eyes more accurate than Adaboost face/eye detector, at a comparable speed. The proposed approach to determine the class-specific threshold further boosts the performance. The proposed system is systematically evaluated on O2FN, AR and CAS-PEAL databases, and compared with many different approaches. The proposed system consistently and significantly reduces the overall equal error rate on all three databases.

3.1 Introduction

Mobile devices have been widely utilized in our daily life. A large amount of private information is stored on mobile devices, e.g. personal videos, pictures, documents or even financial transaction records. Thus, it imposes high security concerns on mobile devices. Traditionally, mobile devices are secured by password. Short password is easily attacked and long password is

¹This work is published in International Conference in Computer Vision 2011, Workshop on Mobile Vision (conference paper [3] in author's publication list) and Pattern Recognition (journal paper [1] in author's publication list).

difficult to remember. In contrast, biometrics can not be lost, stolen, shared, forgotten or easily duplicated. Particularly, face recognition has several advantages over others, e.g. it is natural, non-intrusive and easy to use. Fingerprint authentication system has been successfully implemented on mobile devices. However, it may be uncomfortable to the user to release fingerprint due to its usage in law force. Face recognition has less such concern. In addition, an embedded sensor is required to capture the fingerprint, whereas no additional hardware is required to capture the face image since currently most mobile phones are equipped with a camera.

The proposed face recognition system consists of three sub-systems: face detection, alignment and verification. It has been shown that the performance of a fully automated face recognition system highly depends on face alignment accuracy [94]. Subspace approaches are widely used in face recognition and superior performance has been demonstrated in [7–9, 11, 12, 25, 28, 95–98]. However, high precision of face alignment is required for subspace face recognition. Although we can provide guidance for the user to position the camera to get the face aligned, it is not an easy task to get the face precisely aligned even for an experienced user. For a complete face recognition system, it is not trivial to build a fast and accurate face/eye detector.

Mobile face detection for verification introduces both challenges and opportunities. On one side, mobile phones have limited computational power and memory compared with PC, and hence it imposes higher requirement on the detection speed. On the other side, as users are trying to pass the verification system, they intend to capture a large face in the self-taken photo. It results in a “close-range” face detection, where the face occupies a significant portion of the image. Different from general face detection, where millions of scanning windows are required, only a limited number of scanning windows are required for mobile face detection. Thus, it imposes lower requirement on the detection speed. The number of scanning windows for eye detection is also limited as eyes are detected within the face window. In fact, many face detection tasks belong to “close-range”, e.g. in face tracking, after the first detection, the face can be detected within a small neighborhood in the subsequent frames.

In this chapter, a subspace face/eye detector is proposed. Subspace approaches have been successfully applied in object classification and have demonstrated impressive classification performance [24, 25]. However, it is not widely used in object detection due to the speed limitation, e.g. it requires large matrix multiplication to derive subspace features. Inspired by Viola & Jones' face detector [13], an attentional cascade structure is proposed for subspace approach to speed up the detection process. Classifiers built from a few subspace features are utilized in the early stages to fast filter out most of the negative samples and only those tough samples are evaluated by classifiers with more subspace features in the latter stages. On O2 XDA Flame, it only takes 0.26 seconds to detect both eyes for the proposed face/eye detector. The proposed subspace face/eye detector is optimized for mobile face verification, which aims to achieve higher detection accuracy at a comparable speed with respect to Adaboost face/eye detector.

For face verification sub-system, a novel approach to determine the class-specific threshold (CST) is proposed, which greatly improves the performance of the system. Naturally, some faces are easy to be recognized and some are difficult. Therefore, a CST in general provides better performance than a global threshold (GT). Traditionally, a separate validation dataset is needed to determine the CST. However, separating a portion of the dataset as the validation dataset results in less training data, and hence the performance may degrade. The proposed approach determines the CST based on the training data only. The experimental results on O2FN, AR [99] and CAS-PEAL [100] databases demonstrate the superior performance of the proposed approach.

In order to evaluate the performance of the proposed system on mobile device, the O2FN database ² is built. It contains 2000 images from 50 persons, each with 40 images. The variations in facial expressions are minimized. It mainly contains illumination variations and in-plane rotations.

²“O2” stands for “O2” XDA Flame; “F” stands for “front” camera; “N” stands for “native” setting, e.g. no advance image processing enabled during capturing.

The proposed system is implemented on O2 XDA Flame, on which it takes 1.03 seconds to recognize a face. Comprehensive experimental results on O2FN, AR [99] and CAS-PEAL [100] databases demonstrate the superior performance of the proposed system compared with the system built upon Adaboost face/eye detector.

3.2 Related Work

The related work on each sub-system, e.g. face detection, alignment and verification is discussed first, followed by a review of several complete systems.

In [101–105], Adaboost face detector [13] was built on mobile devices. The main advantage of Adaboost face detector is its speed, which is critical in general face detection. Due to the limited discrimination power of rectangular features, a huge amount of samples are required to train a Viola & Jones' face detector, and hence training is time consuming. The proposed face detector detects faces more precisely at a comparable speed with respect to Adaboost face detector on “close-range” face images. In addition, the proposed face detector has the advantages of fast training and easy implementation.

Face alignment is critical to subspace face recognition. The face is often aligned through eye locations. In [101, 104] Adaboost eye detector was built. Tao & Veldhuis extended it to detect more fiducial points [103]. Eyes are detected within the detected face window, and hence the location and scale of eyes are roughly known. Only a limited number of scanning windows are needed. Thus, Adaboost eye detector may not be the best choice due to its limited accuracy.

Although tons of face verification algorithms are proposed in literature, very few are suitable for mobile applications. In [106], superior performance is achieved for SIFT features, but the recognition speed is extremely slow. Sparse representation [95] demonstrated impressive performance on recognizing partially occluded faces. However, due to the huge computational cost on iterative optimization, it is not suitable for mobile applications. Simple fea-

tures, e.g. Local binary pattern (LBP) [103, 104], Haar Wavelet features [107], DCT coefficients [101], Fourier Coefficients [102] and LRP features [108] are extracted for classification on the resource-constraint devices. Correlation filter [105, 109] is another option due to its simplicity and tolerance to noise. However, those approaches do not yield optimal recognition performance. Subspace face recognition approaches have demonstrated superior performance in [7–9, 11, 12, 25, 28, 95–98]. In [110], Yang et al. ported Direct Linear Discriminant Analysis [111] into mobile devices. In this work, various subspace approaches are implemented and tested on the O2FN database. The best performed one is built into our system.

Several complete face verification systems on mobile devices have been proposed in literature. Ng et al. implemented a real-time face verification system on O2 XDA Ili [105], using Viola & Jones' face detector and Unconstrained Minimum Average Correlation Energy (UMACE) filter for face verification. It was tested on a database of 25 subjects and the overall equal error rate (EER) is 8.49%. Schneider et al. implemented FaceStry on Nokia 6630 [112], but they only performed a sampling check on their system. Hadid et al. implemented a system on Nokia N90 [104], using Viola & Jones' face/eye detector and LBP features for face verification. Tao & Veldhuis implemented a system on Eten M500 Pocket PC [103], using Viola & Jones' detector for face/facial landmark detection and a simplified LBP filter for face verification. It was tested on a database of 20 subjects, and the EER was 2% by fusing the results from 450 frames. However, it is not practical to take a video of 30 seconds for authentication. Adaboost face detector [103–105] and Adaboost eye detector [103, 104] are popular because of its fast speed. UMACE filter [105], geometric features [112] and LBP features [103, 104] are used for face verification because of their simplicity. However, they do not yield optimal recognition performance. In the following sections, we will show that the proposed system achieves much higher recognition accuracy at a comparable speed.

3.3 The Proposed Face Verification System on Mobile Devices

The proposed face verification system consists of three sub-systems: face detection, eye detection and face verification as shown in Fig. 3.1. In view of the asymmetric nature of face/eye detection, by studying the eigen-spectrum of the covariance matrices of face/non-face class and eye/non-eye class, a subspace face/eye detector is proposed. An attentional cascade structure is proposed to speed up the detector. Various subspace face recognition algorithms are studied and the one most suitable for mobile devices is built into our system with the proposed mechanism of class-specific decision.

3.3.1 Face Detection

Asymmetric PCA (APCA): Face detection can be formulated as a two-class classification problem: face class and non-face class. Traditionally, PCA treats two classes equally. As pointed in [24, 25], the role of PCA in pattern classification is to remove the unreliable dimensions that

Figure 3.1: Block diagram of the proposed face verification system on mobile devices.

are harmful to accurate classification. Since the training samples of two classes may have different reliability, we should weight them according to the class reliability when applying PCA. The covariance mixture Σ_α is defined as:

$$\Sigma_\alpha = \alpha \Sigma_p + (1 - \alpha) \Sigma_q + \Sigma_m \quad (3.1)$$

where Σ_p, Σ_q are the covariance matrices of face/non-face class; Σ_m is the between-class scatter matrix; α is the weighting factor of Σ_p and $0 \leq \alpha \leq 1$.

Face detection is highly asymmetric in nature. In general face detection, it is relatively easy to collect representative face samples, but not easy to collect representative negative samples, since the non-face is “everything else except face”. However, in “close-range” face detection the non-face samples are well defined, e.g. mainly the cropped partial faces. As a result, Σ_q can be estimated as reliable as Σ_p . Furthermore, much more non-face samples than face samples can be collected, and hence Σ_q can be more reliably estimated. Dimensions corresponding to the small eigenvalues of the covariance matrix of unreliable class should be removed. Therefore, we assign a large weight α to Σ_p . PCA is applied on Σ_α , and the eigenvectors Φ_m corresponding to first m largest eigenvalues are extracted. By assigning a large weight to Σ_p , more dimensions corresponding to the unreliable small eigenvalues of Σ_p are removed, and hence better generalization performance is achieved.

Asymmetric Discriminant Analysis (ADA): APCA is effective to alleviate over-fitting problem by removing the unreliable dimensions of covariance mixture matrix. For a fast classification, discriminant analysis is necessary to extract a compact feature set from the reliable APCA subspace. Linear Discriminant Analysis (LDA) is commonly used to extract discriminant features. However, for two-class classification problem, only one feature can be extracted. Covariance Discriminant Analysis (CDA) [113] can provide more dimensions, but the asymmetric nature of training data reduces the effectiveness of CDA. In fact, these two approaches can be combined in one discriminant evaluation. In Asymmetric Principal and Discriminant Analysis

(APCDA) algorithm [24], ADA is applied in the APCA subspace, which aims to maximize the Bhattacharyya distance [114] defined in Eqn. (3.2),

$$D = \frac{1}{8}(\hat{M}_p - \hat{M}_q)^T \left(\frac{\hat{\Sigma}_p + \hat{\Sigma}_q}{2} \right)^{-1} (\hat{M}_p - \hat{M}_q) + \frac{1}{2} \ln \frac{|(\hat{\Sigma}_p + \hat{\Sigma}_q)/2|}{\sqrt{|\hat{\Sigma}_p||\hat{\Sigma}_q|}}, \quad (3.2)$$

where $\hat{M}_p, \hat{M}_q, \hat{\Sigma}_p, \hat{\Sigma}_q$ are the mean vectors of the face/non-face class and the covariance matrices of the face/non-face class in APCA subspace, respectively.

The first term of Bhattacharyya distance is maximized by LDA in Eqn. (3.3), where $\hat{\Sigma}_m$ is between-class covariance matrix in APCA subspace.

$$\hat{\Sigma}_m \Phi = (\hat{\Sigma}_p + \hat{\Sigma}_q) \Phi \Lambda \quad (3.3)$$

It is proven in [115] that the second term of Bhattacharyya distance is maximized in the subspace spanned by the generalized eigenvectors corresponding to the largest $\lambda_k + 1/\lambda_k$, where λ_k is the generalized eigenvalue of matrix pair either $(\hat{\Sigma}_p, \hat{\Sigma}_q)$ or $(\hat{\Sigma}_q, \hat{\Sigma}_p)$, or equivalently $(\hat{\Sigma}_p, \hat{\Sigma}_p + \hat{\Sigma}_q)$. However, covariance matrix $\hat{\Sigma}_p, \hat{\Sigma}_q$ are still biased even after APCA, thus the generalized eigenvalue problem is modified as in Eqn. (3.4), where $0.5 < \beta \leq 1$ is determined by prior knowledge about the asymmetry of the two classes.

$$\hat{\Sigma}_p \Phi = (\hat{\Sigma}_p + \beta \hat{\Sigma}_q) \Phi \Lambda \quad (3.4)$$

Then, Eqn. (3.3) and (3.4) are combined, e.g.,

$$(\hat{\Sigma}_p + \gamma \hat{\Sigma}_m) \Phi = (\hat{\Sigma}_p + \beta \hat{\Sigma}_q) \Phi \Lambda, \quad (3.5)$$

where γ is a constant that weights the discriminatory information of the class mean against the covariance. ADA is applied in the APCA subspace and extracts d eigenvectors $\tilde{\Phi}_d$ corresponding to the d largest $\max(\lambda_k, 1 - \lambda_k)$, where λ_k is the generalized eigenvalue of Eqn. (3.5).

Cascade APCDA (C-APCDA) Face Detector: Most non-face samples are significantly dif-

ferent from the face samples. It is possible to filter out a large number of non-face samples and preserve most face samples by utilizing a few APCDA dimensions. Thus, cascade APCDA (C-APCDA) is proposed to boost the detection speed. Easy samples can be processed extremely fast and only those tough samples will be evaluated at the latter stages with more dimensions.

After ADA, feature dimensions are sorted according to their discriminant power in the descending order. The proposed C-APCDA face detector consists of l stages. At stage i , it is an APCDA face detector Ω_i of d_i dimensions, $i = 1, 2, \dots, l$. We choose the first d_i eigenvectors from $\tilde{\Phi}_d$ to construct the projection matrix $U_i = [\tilde{\Phi}_1 \tilde{\Phi}_2 \dots \tilde{\Phi}_{d_i}]$, where $\tilde{\Phi}_i$ is i -th eigenvector of $\tilde{\Phi}_d$. As the stage number increases, the number of dimensions increases, and better classification performance is expected. Here, the face detector at each stage is constructed from the first d_i eigenvectors, as these feature dimensions are un-correlated. For each scanning window, we define its score [24] as in Eqn. (3.6):

$$s_i = (\tilde{X}_i - \tilde{M}_q)^T (\beta \tilde{\Sigma}_q)^{-1} (\tilde{X}_i - \tilde{M}_q) - (\tilde{X}_i - \tilde{M}_p)^T \tilde{\Sigma}_p^{-1} (\tilde{X}_i - \tilde{M}_p) \quad (3.6)$$

where $\tilde{X}_i$, $\tilde{M}_p$, $\tilde{M}_q$, $\tilde{\Sigma}_p$, $\tilde{\Sigma}_q$ are testing sample, the mean vector of face/non-face class, the covariance matrix of face/non-face class in the APCDA subspace, respectively. If $s_i \geq T_j$, where T_j is the threshold at j -stage, it is a face sample, otherwise it is a non-face sample. The optimal threshold at each stage is determined as follows: assume the target false rejection rate (FRR) is ϵ_F , and each stage has equal FRR ϵ , then ϵ can be estimated by $(1 - \epsilon)^l = 1 - \epsilon_F$. The threshold at each stage is adjusted accordingly to achieve desired ϵ . As there is only one face in the image, at the last stage Ω_l , the scanning window with maximum score is detected as the face window.

The procedures to build the proposed C-APCDA face detector are summarized as follows:

1. Perform APCA on Σ_α as defined in Eqn. (7.23) to remove the unreliable dimensions.
2. Perform ADA in the APCA subspace, and choose generalized eigenvectors $\tilde{\Phi}_d$ of Eqn. (3.5) corresponding to d largest $\max(\lambda_k, 1 - \lambda_k)$.

3. Select the first d_i APCDA dimensions to construct the face detector Ω_i at i -stage of the C-APCDA face detector.
4. At the last stage Ω_l , the scanning window with maximum score as defined in Eqn. (3.6) is detected as the face window.

3.3.2 Eye Detection

Although the eye detector can be built in a similar approach to the C-APCDA face detector, there are some differences between these two. Eye detection is to precisely locate two eyes, while face detection is to roughly locate the face region. Eye localization should be as accurate as possible so that the face can be aligned precisely, whereas much larger margin is allowed for face detection. For eye detection, only the cropping window at the precise eye location can be used as an eye sample. The scale of eyes is roughly known since eyes are detected within the detected face window. Therefore, only one eye sample can be generated for each eye. In order to determine the precise eye locations, the scanning windows that are very close to the eyes are included in the non-eye database. Lastly, eye detection is more sensitive to obstacles, e.g. wearing or not wearing glasses is not critical to face detection, but it is critical to eye detection. Similarly, open eyes, half-open eyes or closed eyes greatly affect the appearance of the eyes. All these increase the intra-class variations.

Eye detection is modeled as a two-class classification problem, e.g. positive eye class and negative non-eye class. The non-eye is not “everything else except eye”, but the scanning windows within the face region. Thus, the covariance matrix of the non-eye class can be estimated as reliably as that of the eye class. Furthermore, much more non-eye samples can be collected, and hence the covariance matrix of the non-eye class can be estimated more reliably. A larger weight should be put on the less reliable covariance matrix of the eye class so that more unreliable dimensions from the eye class can be removed.

The face is assumed to be almost upright. Thus, the left eye is detected in the left-up quarter and the right eye is detected in the right-up quarter. Right eyes are left-right flipped and only the left eye detector is built. The procedures similar to the C-APCDA face detector are adopted to build the proposed C-APCDA eye detector.

3.3.3 Face Verification

Many face recognition algorithms have been proposed in literature. Due to the limited resources on mobile devices, many are not suitable for mobile applications. In contrast, subspace approaches require less memory and computational power, and work well on low-resolution images. Thus, they are more suitable for mobile applications. Subspace approaches have demonstrated state-of-the-art performance in face recognition [7–9, 11, 12, 25, 28]. Our face verification sub-system is based on subspace approaches.

Subspace face recognition can be formulated as global formulation or class-specific formulation. In global formulation, one unified projection matrix is trained for all the classes through constructing a pooled covariance matrix [116]. In class-specific formulation, a class-specific projection matrix is trained for each class [117]. Similar to the training process, to determine whether a user is a genuine user or an imposter user, either a global threshold (GT) or a class-specific threshold is needed. Now the question is: *Which combination of formulation and threshold should be used for face verification?* In class-specific formulation, the genuine model generated from the limited number of training samples of each class may not be able to fully capture high complexity of face manifold, and may be easily over-fit to the training data. Those small eigenvalues of the class-specific covariance matrix are largely deviated from their true values [24, 25]. The inverses of those small unreliable eigenvalues are used to weight the features, which causes severe over-fitting problem. In contrast, in global formulation the within-class scatter matrix is constructed by summing up the class-specific covariance matrix of each class, and hence it can be better estimated. As a result, better generalization performance is

achieved for global formulation. For class-specific formulation, n^2 (n is face dimensionality) free parameters of the class-specific covariance matrix need to be estimated from very few training samples of each class, whereas only one parameter needs to be estimated for the CST. Thus, CST will not cause severe over-fitting. Naturally, some faces are easy to be recognized, and hence a small threshold will be sufficient, whereas others may need a large threshold. Thus, CST in general provides better performance than GT. Therefore, we propose to train the face verification system based on global formulation and make the decision based on CST. In the latter section, a novel approach is proposed to determine the CST without sacrificing part of the training data as the validation data, which further boosts the performance of the system.

Subspace approaches share the same framework, e.g. originated from Linear Discriminant Analysis. They aim to find the most discriminant dimensions by maximizing the Fisher ratio. Face dimensionality n is often larger than the number of samples N . $\mathbf{S}_w \in \mathbb{R}^{n \times n}$ is singular since its rank $r \leq N - c$, where c is the number of class. Many approaches are proposed to solve this singularity problem. Fisherface [7] solves the singularity problem of $\mathbf{S}_w$ by applying PCA on $\mathbf{S}_t$ to reduce the feature dimensionality to $N - c$. However, discriminant information also resides in those dimensions being thrown away. Null space approach (NLDA) [8] only utilizes the null space of $\mathbf{S}_w$, but the discriminant information in range space is ignored. In fact, discriminant information resides in both subspaces. The features in the range space can be weighted by their discriminant power, e.g. $\phi_k^w / \sqrt{\lambda_k^w}$, where ϕ_k^w, λ_k^w are k -th eigenvector and eigenvalue of $\mathbf{S}_w$. However, for the features in the null space, $\lambda_k^w = 0$. It is not feasible to directly combine the features from those two subspaces. In addition, in the range space, only the eigenvectors corresponding to leading largest eigenvalues are reliable. Those dimensions corresponding to small eigenvalues are not reliable, but a large amount of discriminant information resides in those dimensions.

In Eigenfeature Regularization and Extraction (ERE) approach [12, 25], it not only alleviates the problem of unreliable small and zero eigenvalues caused by noise, but also enables

discriminant evaluation to be performed in the full dimension of the image data under one unified discriminant criteria. The subspace spanned by eigenvector ϕ_k^w of S_w is decomposed into three subspaces: reliable range space $\mathbf{F} = \{\phi_k^w\}_{k=1}^m$, unreliable range space $\mathbf{N} = \{\phi_k^w\}_{k=m+1}^r$ and null space $\mathbf{\emptyset} = \{\phi_k^w\}_{k=r+1}^n$, where r is the rank of S_w as defined early, and dimension ϕ_{m+1}^w is the starting dimension of unreliable range space. The face component in $\mathbf{F}$ decays rapidly and can be viewed as the outliers of the whole spectrum. Median operator works well in separating outliers from a dataset. To determine the start point of unreliable range space, we first find a point near the center of unreliable range space by $\lambda_{med}^w = \text{median}\{\forall \lambda_k^w | k \leq r\}$. The distance between λ_k^w and the smallest nonzero eigenvalue is $d_{m,r} = \lambda_{med}^w - \lambda_r^w$. The upper bound of the unreliable eigenvalues is estimated by $\lambda_{med}^w + d_{m,r}$. More generally, λ_{m+1}^w can be estimated by:

$$\lambda_{m+1}^w = \max\{\forall \lambda_k^w | \lambda_k^w < \lambda_{med}^w + \mu(\lambda_{med}^w - \lambda_r^w)\}, \quad (3.7)$$

where μ is a constant and $\mu \approx 1$. The optimal μ may vary for different applications. For simplicity, $\mu = 1$ is used for all the experiments.

In the reliable range space $\mathbf{F}$, the eigenvalues remain unchanged. In the unreliable range space $\mathbf{N}$, the new eigenvalues are determined by fitting the eigenvalues in the reliable range space $\mathbf{F}$ into a nonlinear model, e.g. $\tilde{\lambda}_k^w = \frac{\alpha}{k+\beta}$. Although not limited from other possible fitting, α and β are determined by letting $\tilde{\lambda}_1^w = \lambda_1^w$ and $\tilde{\lambda}_m^w = \lambda_m^w$, which yields $\alpha = \frac{\lambda_1^w \lambda_m^w (m-1)}{\lambda_1^w - \lambda_m^w}$, $\beta = \frac{m\lambda_m^w - \lambda_1^w}{\lambda_1^w - \lambda_m^w}$. As a result, the over-fitting problem caused by those unreliable dimensions is alleviated. In the null space $\mathbf{\emptyset}$, the new eigenvalues are chosen as the first eigenvalue of the nonlinear model in the null space $\mathbf{\emptyset}$. As a result, the singularity problem of S_w is alleviated. In summary, the eigen-spectrum λ_k^w is regularized by an eigen-spectrum model:

$$\tilde{\lambda}_k^w = \begin{cases} \lambda_k^w & \text{if } k \leq m \\ \frac{\alpha}{k+\beta} & \text{if } m < k \leq r \\ \frac{\alpha}{r+1+\beta} & \text{if } r < k \leq n \end{cases} \quad (3.8)$$

Discriminant information is extracted in the full image dimension under one unified discriminant criteria. Now the features are weighted by $\tilde{\lambda}_k^w$ as:

$$\tilde{\Phi}_w = \{\phi_1^w / \sqrt{\tilde{\lambda}_1^w}, \phi_2^w / \sqrt{\tilde{\lambda}_2^w}, \dots, \phi_n^w / \sqrt{\tilde{\lambda}_n^w}\} \quad (3.9)$$

Different eigen-spectrum models can be built. The optimal model is application-dependent. For simplicity, the original model in [12] is used, as no free parameter selection is needed.

Then, discriminant analysis is applied to extract a compact feature set. The between-class scatter matrix S_b is projected as: $\tilde{S}_b = \tilde{\Phi}_w^T S_b \tilde{\Phi}_w$. Then, we perform eigen-decomposition on $\tilde{S}_b$, i.e. $\tilde{S}_b = \tilde{\Phi}_b \tilde{\Lambda}_b \tilde{\Phi}_b^T$. The first d eigenvectors $\tilde{\Phi}_b^d$ corresponding to d largest eigenvalues of $\tilde{S}_b$ are selected. Finally, the projection matrix is obtained as: $\tilde{P} = \tilde{\Phi}_w \tilde{\Phi}_b^d$.

The procedures of ERE approach are summarized in Fig. 3.2.

Class-Specific Decision Based on Training Data: Conventionally, a validation set is needed to find the optimal threshold of each class. Separating a validation set from the training set results in less training samples, and hence the overall performance of the system may degrade.

Figure 3.2: Block diagram of ERE approach.

Instead, we propose to determine the CST based on the training data.

The proposed approach is based on the following observation: for the imposter classes whose training samples are very close to the claimed class center, their testing samples are also likely close to the claimed class center, and hence easily falsely accepted. In view of this, we can use the distances from the claimed class center to other class centers as the indicator of the CST, e.g. $d_{i,j} = D(M_i, M_j), i \neq j$, where M_i is the claimed class center, M_j is the class center of j -th class, and D is a distance measure. We can choose the distances to k -nearest class centers and use the average as the indicator of the CST. For simplicity, we choose $k = 1$. The illustration for determining the CST based on the training data is shown in Fig. 3.3. Imposter samples that are close to the claimed class center are easily falsely accepted. Thus, the distance from the claimed class center to its nearest class center is chosen as the indicator of the CST, i.e. $d_i = \min_j(d_{i,j})$. As shown in Fig. 3.3, $d_{1,2}$, $d_{2,3}$, $d_{3,2}$ and $d_{4,1}$ are the indicators for Class 1, 2, 3, 4, respectively. Based on this distance, the CST is calculated as $t_i = d_i R$, where $R \geq 0$ is used to control the security level of the system. If a high security level is required for some applications, i.e. a low false acceptable rate is desirable, user can choose a small value of R . If a low false rejection rate is desirable, user can choose a large value of R . In this way, the proposed approach determines the CST without a validation dataset. More samples can be used in training, and hence a better performance is expected.

In literature, to the best of our knowledge, when the CST is used, no Receiver Operating Characteristic (ROC) curve is provided except some discrete operational points. When a validation dataset is used to determine the CST, the number of operational points is determined by the limited number of genuine validation samples in each class. Therefore, only a limited number of operational points can be determined. For the proposed approach, ROC can be easily generated by varying the control parameter R .

In order to determine the CST, a large amount of training samples of each user are required. For general face recognition tasks, a large number of users use a public system. Thus, it is very

Figure 3.3: Illustration for determining the class-specific threshold based on the training data. The small stars, circles, squares and diamonds represent samples of Class 1-4, respectively. The big ones represent the class centers. The distance of a class center to its nearest class center is used as the indicator of the CST.

difficult to collect such a large amount of training images per each user. However, for mobile applications, the mobile device is personalized. Thus, a user can comfortably collect a large number of his/her own photos gradually during the usage of the system, which can be used to compute a reliable CST.

3.4 O2FN Mobile Face Database Creation

In order to evaluate the proposed face verification system, the O2FN mobile face database is built. It contains 2000 face images of size 144×176 pixels from 50 subjects. The images are self-taken photos. The users are told to take roughly 20 indoor images and 20 outdoor images, with minimum facial expression variations and out-plane rotations. As shown in Fig. 3.4, the O2FN database mainly contains in-plane rotations and illumination variations.

The O2FN database is then used to generate the training and testing databases to evaluate each sub-system. For the face detection sub-system, 53924 face samples and 477097 non-face

Figure 3.4: Sample images of the O2FN database.

samples are generated and normalized to the size of 24×28 pixels³. For the eye detection sub-system, the detected faces are first normalized to 96×112 pixels, and then 4000 eye samples and 207331 non-eye samples of size 35×25 pixels are generated from the detected faces. To evaluate the face verification sub-system, face images are cropped, rotated to the same horizon and normalized to the size of 82×96 pixels based on the manually marked eyes. In the system-level evaluation, face images are normalized based on the eye locations determined by face/eye detector. An ellipse mask is applied to remove the unwanted corner, and the final face vector is of length 6634. Histogram equalization is applied to reduce illumination variations and the face vector is normalized to zero mean and unit variance.

³Face/nonface samples are generated by scanning the images using windows of different sizes. If the scanning window is overlapped with the ground-truth face window by more than 85%, it is a face sample; otherwise it is a non-face sample

3.5 Experimental Results

In this section, the proposed face verification system is firstly systematically evaluated on the O2FN database. Then, it is further compared with the system built upon Adaboost face/eye detector on the AR database [99] and the CAS-PEAL database [100]. Superior performance of the proposed system is demonstrated.

The O2FN database is partitioned in a consistent way as shown in Table 3.1. The performance on each sub-system is evaluated in a pilot run, where 20 odd images of the first 40 subjects are used in training. Then, in the system-level evaluation, the experiment is repeated 5 times. For each trail, 40 subjects are randomly selected and used in training.

Table 3.1: Partition of the O2FN database

	40 subjects		10 subjects
	20 odd images	20 even images	40 images
Face Detection	Training	Testing	Testing
Eye Detection	Training	Testing	Testing
Face Verification	Training	Genuine matching	Imposter matching

3.5.1 Face Detection Results on the O2FN Database

In this section we show the gradual performance improvement, i.e. PCA $\rightarrow$ APCA $\rightarrow$ APCDA $\rightarrow$ C-APCDA face detector on the O2FN database. Then, the proposed C-APCDA face detector is compared with Adaboost face detector.

To study the reliability of face/non-face training samples, we analyze the eigen-spectrums of Σ_p and Σ_q . We perform eigen-decomposition on Σ_p, Σ_q , and obtain corresponding eigenvalue λ_k^p, λ_k^q and eigenvector Φ_k^p, Φ_k^q . Then, face/non-face testing samples are projected onto Φ_k^p, Φ_k^q , respectively. Corresponding variance v_k^p, v_k^q are calculated. We present λ_k^p, v_k^p in the descending order, and λ_k^q, v_k^q in the ascending order as shown in Fig. 3.5(a). λ_k^q and v_k^q are almost overlapped, and hence non-face training samples can well represent testing samples. On the other hand,

Figure 3.5: Experimental results on the O2FN database. (a) Eigen-spectrum plot for the eigenvalue λ_k^p, λ_k^q of the face/non-face covariance matrix, and the corresponding variance v_k^p, v_k^q of the face/non-face testing samples. (b) The EER for the PCA/APCA face detector. (c) The EER for the APCDA face detector.

λ_k^p and v_k^p differ significantly after dimension 300, and hence face training samples cannot well represent face testing samples. A large weight $\alpha = 0.9$ is assigned to Σ_p so that more unreliable dimensions from the face class can be removed by APCA. When $\alpha = n_p/(n_p + n_q) \approx 0.1$, it is a standard PCA face detector. n_p, n_q are the number of face/non-face training samples.

Then, we test the PCA/APCA face detector on extracted testing samples, and plot the EER v.s. dimensions as shown in Fig. 3.5(b). APCA consistently outperforms PCA after dimension 60. It is consistent with our analysis in Section 3.3.1. By assigning a large weight to the

unreliable face class, more dimensions from the unreliable covariance matrix of the face class are removed. Thus, better generalization performance is achieved.

The objective of ADA in APCDA is to find the most discriminant dimensions within the APCA subspace. The EER of the APCDA face detector of different dimensions is shown in Fig. 3.5(c). The EER of the APCDA face detector is comparable to that of the APCA face detector, but with much less dimensions. With 60 dimensions, the lowest EER of 2.73% is achieved for the APCDA face detector. Even with the first dimension only, the EER of 5.32% is achieved.

In view of the discriminant power of first few APCDA dimensions, a cascade APCDA face detector is built. It consists of 8 stages with 1, 2, 3, 6, 12, 24, 48, 96 dimensions at each stage. A large amount of negative samples are filtered out by the classifier built from the first APCDA dimension. The proposed C-APCDA face detector is tested on 1200 face testing images. Let us define the overlap ratio R_O as the ratio of the overlapped region over the union area. 97.00% and 96.50% of faces are detected for $R_O \geq 85\%$ when $\epsilon = 0.5\%$ and $\epsilon = 1\%$, respectively. ϵ is false rejection rate at each stage when training the cascaded APCDA face detector. For $R_O \geq 70\%$, almost perfect detection rate is achieved: 99.67% and 99.25%, respectively. Unless otherwise state, the face detection rate is calculated based on $R_O \geq 70\%$. In terms of time saving, the proposed C-APCDA face detector with $\epsilon = 0.5\%$ is 5.7 times faster than the APCDA face detector of 96 dimensions.

The proposed face detector is further compared with Adaboost face detector. As it requires a huge number of samples and tremendous time to train an optimal Adaboost face detector, we utilize pre-trained face detectors in OpenCV library [118–121]. We test four Adaboost face detectors⁴ and the one with best performance (haarcascade_frontalface_alt_tree.xml [119]) is reported. Table 3.2 summarizes the experimental results of the APCA face detector of 300 di-

⁴Four face detectors are haarcascade_frontalface_default.xml [119], haarcascade_frontalface_alt.xml [119], haarcascade_frontalface_alt_tree.xml [119] and minhtri_frontalface.xml [120, 121].

mensions, the APCDA face detector of 100 dimensions, the proposed C-APCDA face detector and Adaboost face detector. The number of dimensions required for the C-APCDA face detector is estimated as $96/5.7 \approx 17$. Among three subspace face detectors, the C-APCDA face detector needs much less dimensions compared with the other two for roughly the same detection rate. Compared with Adaboost face detector, the C-APCDA face detector achieves better face detection rate.

Table 3.2: Comparisons of APCA, APCDA, C-APCDA face detector and Adaboost face detector on the O2FN database

	$R_O \geq 85\%$	$R_O \geq 70\%$	Dimensions Required
APCA300	97.25%	99.58%	300
APCDA100	97.00%	99.83%	100
C-APCDA	97.00%	99.67%	17
Adaboost Face Detector	96.17%	99.17%	-

3.5.2 Eye Detection Results on the O2FN Database

In this section, we present the experimental results of the APCA/APCDA eye detector, the proposed C-APCDA eye detector and the comparisons to other eye detectors. Eyes are detected within the detected face region. Thus, the detection rate reported here is the overall detection rate of face/eye detection.

To study the reliability of eye/non-eye training samples, we apply eigen-spectrum analysis similarly as in the face detection. For eye samples, eigenvalue λ_k^p of the covariance matrix Σ_p and corresponding variance v_k^p of the testing samples are presented in the descending order as shown in Fig. 3.6(a). For non-eye samples, eigenvalue λ_k^q and corresponding variance v_k^q are presented in the ascending order. v_k^p differs significantly from λ_k^p for $k > 350$, whereas v_k^q in general is very close to λ_k^q . A limited number of eye training samples cannot well represent the testing samples, and hence are less reliable. A large weight is assigned to the eye class, e.g. $\alpha = 0.3$. It is significantly larger than the weight in PCA, e.g. when $\alpha \approx 0.02$, APCA is deduced to standard PCA.

Figure 3.6: Experimental results on the O2FN database. (a) Eigen-spectrum plot for the eigenvalue λ_k^p , λ_k^q of the eye/non-eye covariance matrix, and the corresponding variance v_k^p , v_k^q of the eye/non-eye testing samples. (b) The comparison of the PCA/APCA eye detector in terms of EER. (c) The EER of the APCDA eye detector.

Then, the PCA eye detector and the APCA eye detector are tested on extracted samples. The results are shown in Fig. 3.6(b). The APCA eye detector consistently outperforms the PCA eye detector. By assigning a large weight to the unreliable eye class, more dimensions from the unreliable covariance matrix of the eye class are removed. Thus, better performance is achieved.

With much less dimensions, the APCDA eye detector can achieve comparable performance to the APCA eye detector. The EER of APCDA eye detector of different dimensions is shown in Fig. 3.6(c). With first 2 dimensions only, almost the same EER as APCA60 is achieved.

In view of the discriminant power of the first few APCDA features, a C-APCDA eye detector is built. It consists of 6 stages, with 1, 2, 3, 6, 12, 20 dimensions at each stage. It is tested on 1200 face images for $\epsilon = 0.5\%$ and 1% . 97.75% and 98.00% eyes are detected within 5 pixels to the manually marked eyes, respectively. Table 3.3 shows the detection rates under different tolerances. A large portion of eyes are detected within 2-pixel to the manually marked eyes. In the latter sections, unless otherwise states, the eye detection rate is calculated based on 5-pixel tolerance. We test four Adaboost eye detectors ⁵, and the best performed one (haarcascade_mcs_lefteye.xml [124]) is reported. Compared with Adaboost eye detector [118, 124], the proposed C-APCDA eye detector detects the eyes at much more precise locations.

Table 3.3: Eye detection rate on the O2FN database for the C-APCDA eye detector of $\epsilon = 0.5\%$, 1% and Adaboost eye detector on the detected face images under different tolerances T , where T is the tolerance of the detected eye differed from the ground-truth eye location.

	C-APCDA Eye Detector $\epsilon = 0.5\%$	C-APCDA Eye Detector $\epsilon = 1\%$	Adaboost Eye Detector
$T = 1$	53.33%	53.67%	37.33%
$T = 2$	83.25%	84.00%	77.42%
$T = 3$	93.17%	93.42%	90.33%
$T = 4$	96.42%	96.50%	94.25%
$T = 5$	97.75%	98.00%	95.92%

For the C-APCDA eye detector, by utilizing the focus-attention strategy, most of non-eye samples are rejected at the early stages and only those tough samples are evaluated at the latter stages. Thus, the speed is greatly improved. The computational complexity of the C-APCDA eye detector is only 22.82% of the APCDA eye detector of 20 dimensions. Table 3.4 shows the performance comparison of the APCA eye detector of 60 dimensions, the APCDA eye detector of 20 dimensions and the proposed C-APCDA eye detector. For the C-APCDA eye detector, the number of dimensions required is calculated as $20 \times 22.82\% \approx 4.6$. Compared with the APCA/APCDA eye detector, the proposed C-APCDA eye detector achieves a comparable

⁵Four eye detectors are haarcascade_eye.xml [122], haarcascade_eye_tree_eyeglasses.xml [122], haarcascade_lefteye_2splits.xml [123], and haarcascade_mcs_lefteye.xml [124].

detection accuracy, but with much less dimensions.

Table 3.4: Performance comparison of APCA, APCDA and C-APCDA eye detector on the O2FN database

Eye Detector	Detection Rate	Dimension Required
APCA60	97.58%	60
APCDA20	96.83%	20
C-APCDA	98.00%	4.6

The proposed C-APCDA eye detector is further compared with four feature-based eye detectors [125–128] and Adaboost eye detector [118]. Three sets of results for the proposed C-APCDA eye detector are presented: corresponding to scanning step size of 1, 2 and 3. For Adaboost eye detector, the experimental results of Scale Factor 1.01 and 1.1 are both given. The algorithms are implemented and tested on a PC with Intel Core 2 Duo CPU @2.66 GHz and 2GB RAM. The experimental results are shown in Table 3.5. Feature-based eye detectors are simple and fast, and hence suitable for mobile devices with limited computation power. However, as shown in Table 3.5, feature-based eye detectors can not work well. The highest detection rate is achieved for the proposed C-APCDA eye detector of scanning step size 1. The processing time is 166.6 ms, which is longer than Adaboost eye detector of Scale Factor 1.1. As shown in the last two rows of Table 3.5, if we increase the scanning step size of the proposed algorithm to 2 or 3, the proposed C-APCDA eye detector consumes less processing time yet still achieves significantly higher detection rate than Adaboost eye detector. In the latter sections, the Scale Factor of Adaboost eye detector is set as 1.01. Some sample images of face/eye detection results for the proposed C-APCDA face/eye detector are shown in Fig. 3.7. The eyes are detected almost as accurate as the marked eyes.

3.5.3 Face Verification Results on the O2FN Database

ERE approach [12, 25] and various other subspace face recognition algorithms, e.g. Bayesian Maximum Likelihood (BML) [28], Enhanced Maximum Likelihood (EML) [11], Dual-space

Table 3.5: Comparisons of the C-APCDA eye detector with other eye detectors on the O2FN database

Algorithm	Detection Rate	Time (ms)
Li et al. Algorithm [125]	55.63%	2.8
Lin & Yang’s Algorithm [126]	74.17%	27.4
Ren & Jiang’s Algorithm [127]	83.42%	51.8
Ren & Jiang’s Algorithm [128]	89.50%	10.0
Adaboost [118] - SF 1.01	95.92%	576.9
Adaboost [118] - SF 1.1	90.58%	61.0
The proposed Algorithm - Step Size 1	98.00%	166.6
The proposed Algorithm - Step Size 2	97.92%	44.0
The proposed Algorithm - Step Size 3	96.50%	22.6

Figure 3.7: The sample images of face/eye detection results for the C-APCDA face/eye detector on the O2FN database.

LDA (DSL) [9], FisherFace [7], Null space approach [8] and Graph Embedding [98], are implemented and tested on the O2FN database. They are tested on the face images normalized based on the manually marked eyes (“Marked Eyes”) and the detected eyes (“Detected Eyes”), and evaluated for CST and GT. “Detected Eyes” are determined by the proposed C-APCDA face/eye detector. The experimental results at optimal parameters are reported in Table 3.6.

In original FisherFace approach the null space of Σ_w may not be removed when PCA is performed on Σ_t . Those dimensions corresponding to the null space of Σ_w will have very large weights, and hence dominate the matching score. Therefore, we modify original FisherFace by performing PCA on Σ_w to reduce the dimension to $N - c$. Much better performance is achieved. For Graph Embedding, Marginal Fisher Analysis [98] is implemented.

Table 3.6 shows that the ERE approach achieves the best performance for almost all the

Table 3.6: Comparisons of various subspace face recognition algorithms at different settings in terms of EER on the O2FN database.

Algorithm	Marked Eyes		Detected Eyes	
	GT	CST	GT	CST
BML [28]	12.16%	13.52%	16.03%	17.76%
EML [11]	9.63%	11.27%	13.07%	15.23%
Dual-Space [9]	3.13%	1.83%	8.08%	4.98%
FisherFace [7]	32.53%	35.37%	34.43%	37.91%
FisherFace - Modified	2.69%	1.88%	5.87%	4.88%
Null Space [8]	3.16%	1.90%	8.24%	4.98%
Graph Embedding [98]	5.95%	3.36%	11.64%	10.76%
ERE Approach [12]	2.42%	1.37%	6.28%	4.12%

settings. The EER is as low as 1.37% for “Marked Eyes” and CST. When tested for “Detected Eyes”, the EER is as low as 4.12% for CST. The eigen-spectrum model of ERE approach alleviates the over-fitting problem caused by the small eigenvalues in the range space of Σ_w , and utilizes the discriminant information in both the range place and the null space. Thus, better performance is achieved.

Table 3.6 also shows that better performance is achieved for CST compared with GT. The CST determined by the proposed approach greatly improves the performance. In addition, we can easily obtain the ROC for different algorithms. E.g., the ROC for ERE approach, “Marked Eyes”, CST is shown in Fig. 3.8.

Finally, Table 3.6 shows that subspace approaches are sensitive to face alignment error. Previously, we achieve face/eye detection rate as high as 98%, but the performance based on the detected eyes still degrades significantly. Thus, an accurate face/eye detector is crucial to the performance of the complete and fully automated face verification system.

Face recognition via sparse representation [95] has demonstrated superior performance to handle occluded faces. We implement it based on l1-MAGIC [129] and compare it with ERE approach on the images normalized based on the manually marked eyes. The original normalized image of size 82×96 pixels is too large for sparse representation. Thus, we resize it to

41×48 pixels. Sparse representation is originally formulated as face identification problem with a reject option. Sparsity Concentration Index (SCI) is used to reject the imposter samples. Then for the genuine samples, Algorithm 1 in [95] is used to find the identity. To handle the occlusion, an identity matrix is appended after the image matrix. The EER for SCI is 2.94%, and subsequent face recognition rate on the genuine samples is 98.25%. The overall EER of sparse coding will surely be larger than 2.94%. On the same image set of size 41×48 pixels, the EER of ERE approach is 2.5% when using the global threshold and 1.78% when using the CST. Clearly ERE approach outperforms sparse coding.

3.5.4 System-Level Comparison on the O2FN Database

The proposed face verification system, i.e. the C-APCDA face/eye detector, ERE face verification with the CST, is compared with the system built from Adaboost face/eye detector, ERE face verification with the global threshold. The O2FN database is partitioned as shown in Table 3.1. The experiment is repeated 5 times, and each time 40 subjects are randomly selected and used in training. The average performance and standard derivation are reported in Table

Figure 3.8: The ROC for ERE approach based on the class-specific threshold

3.7. The performance is fairly consistent over 5 trials. The EER of ERE face verification on the images normalized based on the manually marked eyes is $0.94\% \pm 0.27\%$. The proposed system demonstrates superior performance compared with the competitive system. An overall EER of 3.88% is achieved for the proposed system.

Table 3.7: Performance comparison of the proposed face verification system and the system built upon Adaboost face/eye detector, evaluated on 2000 images of the O2FN mobile face database for 5 trials

	Face Detection Rate	Eye Detection Rate	System Overall EER
C-APCDA face/eye detector + ERE + CST	99.33% $\pm 0.19\%$	96.98% $\pm 0.29\%$	3.88% $\pm 0.27\%$
Adaboost face/eye detector + ERE + GT	98.73% $\pm 0.16\%$	95.40% $\pm 0.18\%$	8.49% $\pm 0.71\%$

3.5.5 Experimental Results on the AR Database

Table 3.8: Performance comparison of the proposed face verification system and the system built upon Adaboost face/eye detector, evaluated on 1050 images of 75 subjects on the AR database

Algorithm	Face Detection Rate		Eye Detection Rate	Marked Eyes		Detected Eyes	
	$R_O \geq 85\%$	$R_O \geq 70\%$		GT	CST	GT	CST
C-APCDA face/eye detector + ERE	96.98%	99.84%	98.73%	2.70%	1.66%	4.54%	1.90%
Adaboost face/eye detector + ERE	91.27%	99.21%	95.71%			7.64%	3.37%

75 subjects with 14 non-occluded images per subject are selected from the AR database [99]. Those images mainly vary in facial expressions. The first 7 images of first 60 subjects serve as the training images, and the remaining 7 images of first 60 subjects serve as the genuine testing images. The 14 images of last 15 subjects serves as the imposter testing images. The C-APCDA face/eye detector is built based on 420 training images and then used to determine the eye locations of the testing samples. The face verification sub-system is based on ERE

approach. The proposed system is compared with the system built upon the Adaboost face/eye detector as shown in Table 3.8. The abbreviations are the same as defined in Table 3.6.

The proposed C-APCDA face detector locates the faces at a much higher precision compared with the Adaboost face detector, e.g. 96.98% of faces are located with $R_O \geq 85\%$. Subsequently, the proposed C-APCDA eye detector detects the eyes at a higher precision compared with Adaboost eye detector, e.g. 98.73% of eyes are detected within 5 pixels to the manually marked eyes. As a result, compared with the system built upon Adaboost face/eye detector, the proposed system achieves much lower EER, e.g. 4.54% for the global threshold and 1.90% for the CST. In addition, it can be seen that the proposed approach to determine the CST significantly improves the performance.

3.5.6 Experimental Results on the CAS-PEAL Database

There are in total 30863 images of 1040 subjects in the CAS-PEAL database, in which 9031 are frontal faces and 21832 are profile faces. We choose the frontal dataset for testing. The CAS-PEAL database is more challenging than the AR database. Besides variations in facial expressions, there are variations in accessory, aging, background, distance and lighting. 2359 images of first 101 subjects are selected. Other subjects are discarded as they have much less images per subject, and hence not suitable for face verification task. The odd images of first 80 subjects serve as the training images, and the even images of first 80 subjects serve as the genuine testing images. The images of last 21 subjects serve as the imposter testing images. The face verification sub-system is based on ERE approach. The proposed system is compared with the system built upon Adaboost face/eye detector as shown in Table 3.9.

The proposed C-APCDA face/eye detector locates the eyes at a much higher precision compared with Adaboost face/eye detector, e.g. the eye detection rate increases from 93.02% to 98.55%. As a result, the proposed system outperforms the system built upon Adaboost face/eye detector, e.g. the overall EER is reduced from 9.30% to 6.60% for the GT. The proposed ap-

Table 3.9: Performance comparison of the proposed face verification system and the system built upon Adaboost face/eye detector, evaluated on 2359 images of 101 subjects on the CAS-PEAL database.

Algorithm	Face Detection Rate		Eye Detection Rate	Marked Eyes		Detected Eyes	
	$R_O \geq 85\%$	$R_O \geq 70\%$		GT	CST	GT	CST
C-APCDA face/eye detector + ERE	94.48%	99.42%	98.55%	4.13%	3.18%	6.60%	5.60%
Adaboost face/eye detector + ERE	93.46%	99.27%	93.02%			9.30%	6.77%

proach to determine the CST further improves the performance. Compared with the GT, the overall EER of the proposed system is reduced from 6.60% to 5.60% for the CST.

3.5.7 Implementation on Mobile Devices

The proposed complete and fully automated face verification system is implemented on O2 XDA Flame, with 2 Mb camera at the rear, one VGA video camera in front, 520MHz IntelR PXA-270 processor, 128 Mb RAM and Windows Mobile 6 operating system. Two techniques are employed to further speed up the system. Firstly, we utilize the focus-attention strategy in the spatial domain for face/eye detection, e.g. checking the face/eye at a coarse scale first and then focus on the region of interest at a fine scale. Secondly, in view of the discriminant power of the APCDA face/eye detector of first few dimensions, we build the C-APCDA face/eye detector based on down-sampled face/eye samples. The face is down-sampled from 24×28 pixels to 6×7 pixels and the eye is down-sampled from 35×25 pixels to 7×5 pixels. Scanning windows are first filtered by those down-sampled C-APCDA detector, and then only those tough samples are evaluated by the C-APCDA detector of higher resolution. In such a way, the detection speed is further improved. On average, it takes 110.5 ms for face detection, 145.1 ms for eye detection and 772.5 ms for face verification. In total, it takes 1028.1 ms to recognize a face on O2 XDA Flame mobile phone. If the system is implemented on the new mobile phones with more memory and computation power, e.g. iPhone 4S, much higher speed is expected.

The core module is implemented in native C++ (no windows API) and packaged in dll. The system can be easily ported into various applications and different platforms. Several applications have been built upon this system, e.g. mobile access control, personalized media consumption and data/folder locker. For more implementation details, readers can refer to [130].

3.6 Summary

In this chapter, a complete and fully automated face verification system is developed to secure the mobile devices. The proposed system is systematically evaluated on O2FN, AR and CAS-PEAL databases. Comprehensive experiments demonstrate the superior performance of the proposed system compared with the state-of-the-art technologies. The best competitive system is built upon Adaboost face/eye detector and ERE approach for face recognition. Compared with this system, the proposed system reduces the overall EER from 8.49% to 3.88% on the O2FN database, from 7.64% to 1.90% on the AR database and from 9.30% to 5.60% on the CAS-PEAL database. The proposed system is implemented on O2 XDA Flame, on which it takes 1.03 second to recognize a face.

The proposed cascade APCDA face/eye detector achieves a better detection rate than Adaboost face/eye detector. The eye detection rate increases from 95.40% to 96.98% on the O2FN database, from 95.71% to 98.73% on the AR database and from 93.02% to 98.55% on the CAS-PEAL database. The eyes are detected almost as accurate as the manually marked eyes. Furthermore, by utilizing the focus-attention strategy, the detection speed is greatly improved. Compared with Adaboost face/eye detector, the proposed C-APCDA eye face/eye detector detects the eyes in a much higher precision at a comparable speed.

Various subspace approaches are evaluated. In ERE approach, the eigen-spectrum of Σ_w is replaced by a spectrum model, which solves the problem of the unreliable small eigenvalues in the principal subspace and utilizes the discriminant information in both principal and null

subspaces. The proposed approach to determine the CST from the training data instead of extra validation data further improves the performance.

Chapter 4

Noise-Resistant Local Binary Pattern with an Embedded Error-Correction Mechanism

Local binary pattern (LBP) is sensitive to noise. Thus, we propose a noise-resistant LBP (NRLBP) to preserve the image local structures in presence of noise¹. The small pixel difference is encoded as *uncertain* bit first and then determined based on the other certain bits of the LBP code. It is widely accepted that most image micro-structures are represented by uniform codes and noisy patterns are represented by non-uniform codes. Thus, we encode the small pixel difference in such a way that only possible uniform codes are formed. This can be viewed as an error-correction mechanism to recover the distorted image patterns. In addition, we propose an extended noise-resistant LBP (ENRLBP) to capture line patterns. The proposed NRLBP and ENRLBP are more resistant to noise than LBP and its variants.

¹This work is published in International Conference on Image Processing 2013 (conference Paper [5] in author's publication list) and IEEE Transactions on Image Processing (journal paper [2] in author's publication list).

4.1 Introduction

Local binary pattern (LBP) operator transforms an image into an array or image of integer labels describing micro-pattern, i.e. pattern formed by a pixel and its immediate neighbors [131]. More specifically, LBP encodes the signs of the pixel differences between a pixel and its neighbouring pixels to a binary code. The histogram of such codes in an image block is commonly used for further analysis. Its popularity arises from the following advantages. Firstly, the exact intensities are discarded, and only the relative intensities with respect to the center are preserved. Thus, LBP is less sensitive to illumination variations. Secondly, by extracting the histogram of micro-patterns in a patch, the exact location information is discarded, and only the patch-wise location information is preserved. Thus, LBP is robust to alignment error. Lastly, LBP features can be extracted efficiently, which enables real-time image analysis.

However, the sensitivity to noise limits the performance of LBP features [1]. LBP and its variants partially solve the noise-sensitive problem. However, they lack a mechanism to recover the corrupted image patterns. In this chapter, we propose a Noise-Resistant LBP (NRLBP) and an Extended Noise-Resistant LBP (ENRLBP) to address this issue.

The signs of pixel differences used to compute LBP and its variants are vulnerable to noise when they are small. Thus, we propose to encode small pixel difference as an *uncertain* bit first and then determine its value based on the other bits of the LBP code. Uniform patterns are more likely to occur compared with non-uniform patterns in natural images [16, 17]. Most image structures are represented by uniform patterns, and non-uniform patterns are most likely caused by noise. Thus, in the proposed NRLBP, we assign the values of *uncertain* bits so as to form uniform patterns. A non-uniform pattern is generated only if no uniform pattern can be formed. As noise may change an uniform pattern into an unstable non-uniform pattern, the proposed NRLBP corrects many distorted non-uniform patterns back to uniform patterns.

For LBP and LTP, line patterns are treated as non-uniform patterns and grouped into the

non-uniform bin. Uniform patterns mainly represent spot, flat region, edge, edge end and corner. A local image is a line pattern if it is a line against the background, as shown in Fig. 4.5. Line patterns may appear less frequently than uniform patterns, but they represent an important group of local primitives for pattern recognition. Thus, we propose an extension set of uniform patterns corresponding to line patterns. Then, we propose extended noise-resistant LBP (ENRLBP). During the encoding process, we assign the values of *uncertain* bits so as to form extended uniform patterns.

To evaluate our approaches, we first inject Gaussian noise and uniform noise of different noise levels on the AR database [132] for face recognition and the Outex dataset [133] for texture recognition. The proposed approaches demonstrate strong resistance to noise compared with LBP/LTP and its variants. The proposed approaches are further compared with LBP/LTP variants for face recognition on the extended Yale database [134, 135] and the O2FN database [136], protein cellular classification on the 2D Hela database and image segmentation on the Outex dataset [133] and also a natural image downloaded from the web. The proposed NRLBP and ENRLBP consistently achieve comparable or better performance compared with LBP/LTP and its variants.

4.2 Related Work

Many approaches have been proposed in literature to tackle the noise-sensitivity problem of LBP feature. We only review the compared approaches in this chapter. In [16], uniform LBP was proposed to reduce the noise in LBP histogram. The noise is suppressed by grouping the non-uniform patterns into one histogram bin. In [29], Liao et al. consider the most frequently occurred pattern in a texture image rather than uniform patterns, as uniform patterns may have a lower occurrence frequency than some non-uniform patterns. Zhou et al. [39] and Fathi et al. [40] proposed to utilize the information in non-uniform patterns, and group patterns

according to a uniformity measure and the number of ones in the LBP codes. Instead of hard-coding the pixel difference, a fuzzy LBP (FLBP) using piecewise linear fuzzy membership function [42, 43] and another using Gaussian-like membership function [44] utilize a probability measure to represent the likelihood of pixel difference to be encoded as 0 or 1. Local ternary pattern (LTP) was proposed in [1] to tackle the image noise in uniform regions. Instead of binary code, the pixel difference is encoded as a 3-valued code according to a threshold t . Then, the ternary code is split into a positive LBP and a negative LBP in order to reduce the dimensionality.

4.3 Noise-Resistant LBP

4.3.1 Problem Analysis of LBP and LTP

Local binary pattern encodes the pixel difference $z_p = i_p - i_c$ between the neighboring pixel i_p and the central pixel i_c . Let $C_{P,R}^B = \overrightarrow{b_{P-1}^B b_{P-2}^B \dots b_1^B b_0^B}$ denote the LBP code of P neighbors at the distance of R to the center pixel. A code is also called a pattern. Let $LBP_{P,R}$ denote such a coding scheme for $C_{P,R}^B$. Each bit is obtained as:

$$b_p^B = \begin{cases} 1 & \text{if } z_p \geq 0, \\ 0 & \text{if } z_p < 0. \end{cases} \quad (4.1)$$

LBP is widely used in many applications because of its simplicity and robustness to illumination variations. However, LBP is sensitive to image noise. In [16], uniform LBP was proposed to capture fundamental image structures and reduce the noise in LBP histogram. The uniformity U is defined as the number of circularly bitwise transitions from 0 to 1 or vice versa. A local binary pattern is $u2$ -uniform or simply called uniform if $U \leq 2$. For example, “11110000” is a uniform pattern as $U = 2$, whereas “01010111” shown in Fig. 4.1(a) is a non-uniform pattern as

$U = 6$. $LBP_{P,R}^{u2}$ indicates a coding and histogram mapping scheme in which $u2$ -uniform LBP codes of P neighbors at the distance of R to the center pixel are utilized. Uniform patterns occur much more frequently than non-uniform patterns in natural images. It has been shown that $LBP_{8,1}^{u2}$ accounts for almost 90% of all patterns for texture images [16] and $LBP_{8,2}^{u2}$ accounts for 90.6% for facial images [17]. The occurrence probabilities of non-uniform patterns are so small that they cannot be reliably estimated [16]. Inclusion of such noisy estimates in the histogram would harm the classification performance. In addition, non-uniform patterns may be caused by the image noise. Therefore, when constructing the histogram, all non-uniform patterns are grouped into one bin. This not only reduces feature dimensionality, but more importantly, the noise due to unreliable estimates of non-uniform patterns is greatly suppressed. The number of patterns is reduced significantly from 2^P to $P(P-1)+3$. For example, $LBP_{8,2}$ consists of 256 patterns whereas $LBP_{8,2}^{u2}$ has only 59 patterns.

Uniform LBP successfully reduces the noise in LBP histogram, but it is still sensitive to image noise. As shown in Fig. 4.1(a), a small noise will cause the pixel difference encoded differently. Ideally such a smooth region should be encoded as “11111111”. Due to the image noise, it is encoded as “01010111” instead. LTP partially solves this problem by encoding the small pixel difference into a third state [1]. Instead of using binary code, each pixel difference is encoded as a 3-valued code. Let $C_{P,R}^T = \overrightarrow{b_{P-1}^T b_{P-2}^T \dots b_1^T b_0^T}$ denote the LTP code of P neighbors at the distance of R to the center pixel and $LTP_{P,R}$ denote such a coding scheme for $C_{P,R}^T$. Each bit is obtained as:

$$b_p^T = \begin{cases} 1 & \text{if } z_p \geq t, \\ 0 & \text{if } |z_p| < t, \\ -1 & \text{if } z_p \leq -t, \end{cases} \quad (4.2)$$

where t is a pre-defined threshold.

Figure 4.1: (a) An example of LBP encoding scheme for the smooth region with small image noise. LBP is sensitive to image noise. (b) An example of LTP encoding process. LTP doubles the number of patterns compared with LBP.

LTP is more resistant to noise. However, the dimensionality of LTP histogram is very large, e.g. $LTP_{8,2}$ exhibits a histogram of $3^8 = 6561$ bins. Thus, in [1], LTP is split into a positive LBP and a negative LBP. Each bit of positive LBP is obtained as:

$$b_p^p = \begin{cases} 1 & \text{if } z_p \geq t, \\ 0 & \text{if } z_p < t. \end{cases} \quad (4.3)$$

Each bit of negative LBP is obtained as:

$$b_p^n = \begin{cases} 0 & \text{if } z_p \leq -t, \\ 1 & \text{if } z_p > -t. \end{cases} \quad (4.4)$$

To show the commonalities and differences among LBP, LTP and the proposed NRLBP clearly, the negative LBP defined here is the complement of the negative LBP defined in [1].

Effectively they achieve the same result for histogram-based comparison. Eventually, LTP is treated as two separate channels of LBP codes: one channel for positive LBP and the other for negative LBP. In general, uniform LTP is used, in which both channels are uniform LBP. This coding scheme is denoted by $LTP_{P,R}^{u2}$. An example of LTP encoding process is shown in Fig. 4.1(b). LTP doubles the number of patterns compared with LBP.

The small pixel difference may be easily distorted by the noise. Both LBP and LTP lack a mechanism to correct the corrupted patterns. The corrupted image patterns are treated without any attempt to recover the underlining local structures. To address this issue, we propose a Noise-Resistant LBP and an Extended Noise-Resistant LBP.

4.3.2 Proposed Noise-Resistant LBP

LBP is sensitive to noise. Even a small noise may change the LBP code significantly. Thus, we propose to encode the small pixel difference as an *uncertain* bit X first and then determine X based on other certain bits of the LBP code. For the pixel difference z_p between the neighboring pixel and the central pixel, we encode it into one of the three states b_p^N as:

$$b_p^N = \begin{cases} 1 & \text{if } z_p \geq t, \\ X & \text{if } |z_p| < t, \\ 0 & \text{if } z_p \leq -t. \end{cases} \quad (4.5)$$

States 1 and 0 represent two strong states where the pixel difference is almost definitely positive and negative, respectively. Noise can unlikely change them from 0 to 1 or from 1 to 0 . State X represents an *uncertain* state where the pixel difference is small. A small pixel difference is vulnerable to noise if we only take its sign. More specifically, noise can easily change its LBP bit from 0 to 1 or vice versa. Therefore, we encode it as an *uncertain* state

regardless its sign.

Then, we constrain the value of the *uncertain* bit into either 0 or 1, represented by a variable $x_i, x_i \in \{0, 1\}$. Let $\mathbf{X} = (x_1, x_2, \dots, x_n)$ denote the vector formed by n variables of a code. $\mathbf{X} \in \{0, 1\}^n$. The *uncertain* code can be represented by $C(\mathbf{X})$ as:

$$\overrightarrow{b_{P-1}^N b_{P-2}^N \dots b_1^N b_0^N} = C(\mathbf{X}). \quad (4.6)$$

Take the *uncertain* code “11X100X0” in Fig. 4.2(a) for illustration. The *uncertain* code $\overrightarrow{11x_2100x_10}$ can be viewed as the function of $\mathbf{X} = \{x_1, x_2\}$.

After we derive the *uncertain* code, we determine the *uncertain* bits based on the values of the other certain bits to form one or more codes of image local structures. Uniform patterns represent local primitives, including spot, flat, edge, edge end and corner. They appear much more often than non-uniform patterns in natural images. Since uniform patterns occur more likely than non-uniform ones, we assign the values of *uncertain* bits $\mathbf{X}$ so as to form possible uniform LBP codes. A non-uniform pattern is generated only if no uniform pattern can be formed. Take Fig. 4.2(b) as an example. We determine the *uncertain* bit of *uncertain* code “11X1X0X0” so as to form only uniform patterns, e.g. “11110000” and “11111000”.

Mathematically, let Φ_u denote the collection of all uniform LBP codes. For $LBP_{8,2}^{u2}$, Φ_u consists of 58 uniform codes. Based on the *uncertain* code $C(\mathbf{X})$, a set of the proposed NRLBP codes are obtained as:

$$S_{NRLBP} = \{C(\mathbf{X}) | \mathbf{X} \in \{0, 1\}^n, C(\mathbf{X}) \in \Phi_u\} \quad (4.7)$$

Now let us construct the histogram of NRLBP for a local image patch. Let m denote the number of elements in S_{NRLBP} . If $m > 0$, the bin corresponding to each element in S_{NRLBP} will be added by $1/m$. After all, all these patterns originate from one *uncertain* code. If $m = 0$,

the non-uniform bin will be added by 1. This process is repeated for every pixel in the patch. Algorithm 1 summarizes the process.

Algorithm 1: Histogram construction of the proposed NRLBP

for Every pixel in a patch **do**

1. Derive the *uncertain* code $C(\mathbf{X})$ as in Eqn. (4.5), (4.6).

3. Search *uncertain* bits $\mathbf{X}$ in the space $\{0, 1\}^n$ so that $C(\mathbf{X})$ forms uniform LBP codes as in Eqn. (4.7).

4. Construct the histogram.

if $m = 0$ **then**

Accumulate the non-uniform bin with 1.

else

Accumulate the bin of each pattern in S_{NRLBP} with $1/m$.

end if

end for

Now we compare the proposed NRLBP with LBP and LTP by several examples. We consider the cases that different number of LBP codes are derived in S_{NRLBP} . Image patterns in Fig. 4.2(a), (b), (c), (d) generate $m = 1, 2, 3, 4$ NRLBP codes, respectively. Fig. 4.2(e) shows an example where no uniform code can be formed for NRLBP. The corresponding LBP code and LTP code are also given. For LTP, the positive LBP and negative LBP are accumulated in two different histograms, whereas for LBP and NRLBP, the codes are accumulated in one histogram.

As noise may change a uniform image pattern into an unstable non-uniform pattern, the proposed NRLBP corrects such a code back to uniform code. As shown in Fig. 4.2(a), the LBP code is “11010010”, which may be distorted by the noise. The proposed NRLBP first derives the *uncertain* code “11X100X0”, and then determine its *uncertain* bits by forming the uniform code “11110000”. This can be viewed as an error-correction mechanism. Note that we only attempt such an error correction on *uncertain* bits. We do not attempt to correct the non-uniform patterns that are resulted from two strong states. Similarly, we can observe such an error-correction process in Fig. 4.2(b), (c), (d). In these three cases, more than one NRLBP

Figure 4.2: Illustration of encoding process of NRLTP and comparison to LBP and LTP. Fig. (a), (b), (c), (d) are corresponding to $m = 1, 2, 3, 4$ resulting NRLBP codes, respectively. Fig. (e) shows an example that no uniform code can be formed. The proposed NRLBP is significantly different from LBP and LTP. Threshold t is chosen as 2 for LTP and NRLBP in this figure.

code is generated.

The proposed NRLBP corrects noisy non-uniform patterns back to uniform pattern. Fig. 4.3 shows the histogram of LBP, LTP and NRLBP for the image shown in Fig. 4.7(c). The threshold t is chosen as 10 for LTP and NRLBP. LTP histogram is the concatenation of positive LBP histogram and negative LBP histogram. The last bin of each histogram is corresponding to non-uniform patterns, and other bins are corresponding to uniform patterns. Clearly, compared with LBP histogram and LTP histogram, non-uniform patterns in NRLBP histogram are reduced significantly from about 35% to about 10% only. The proposed NRLBP corrects a large amount of non-uniform patterns that are corrupted by the noise back to uniform patterns.

The proposed NRLBP is different from LBP and LTP in many other aspects besides the

capability of noise resistance and error-correction. The LBP code is one of the NRLBP code set if it is uniform. The only exception is that the LBP code is non-uniform and is corrected back to uniform code in NRLBP. Compared with LTP, the treatment of *uncertain* state is totally different for NRLBP. For LTP, all *uncertain* bits are set to 0 for positive half and 1 for negative half as shown in Fig. 4.2, whereas for the proposed NRLBP, we do not hurry for a decision of the *uncertain* bits. We treat them as if they could be encoded as 1 and/or 0, and determine their values based on the other bits of the code. Mathematically, for LTP, $\mathbf{X} \in \{0\}^n$ for positive half and $\mathbf{X} \in \{1\}^n$ for negative half, whereas $\mathbf{X} \in \{0, 1\}^n$ for NRLBP. The number of histogram bin is also different. LTP histogram consists of 118 bins, whereas NRLBP histogram only has 59 bins.

For implementation, a look-up table from the *uncertain* code to the feature vector of NRLBP histogram can be pre-computed. Then, the feature vector of local image patch can be easily obtained by summing up the feature vector of each pixel in this image patch.

Figure 4.3: The histogram of LBP, LTP and NRLBP for the image shown in Fig. 4.7(c). LTP histogram is the concatenation of positive LBP histogram and negative LBP histogram. The last bin of each histogram is corresponding to non-uniform patterns. Compared with LBP histogram and LTP histogram, NRLBP significantly reduces non-uniform patterns from about 35% to about 10%. The proposed NRLBP corrects a large amount of noisy non-uniform patterns back to uniform patterns.

4.3.3 Proposed Extended Noise-Resistant LBP

The local primitives represented by uniform LBP mainly consists of spots, flat region, edges, edge ends and corners [131], as shown in Fig. 4.4. However, a large group of local primitives are totally discarded, e.g. lines patterns, as shown in Fig. 4.5. Although those patterns may not appear as frequently as uniform patterns, they represent an important group of local primitives that may be crucial for recognition tasks. Grouping them with other non-uniform patterns into one bin may result in information lost. Therefore, we introduce an extended set of uniform patterns to preserve line patterns. Among all possible line patterns, diagonal lines appear less frequently. In order to keep the feature vector compact, we only choose nearly horizontal or vertical lines.

Let α denote the angle of the line away from the horizontal line. If $\alpha \in [0, 30^\circ)$ or $\alpha \in (150^\circ, 180^\circ]$, it is considered as a horizontal line. If $\alpha \in [60^\circ, 120^\circ]$, it is considered as a vertical line. If $\alpha \in [30^\circ, 60^\circ)$ or $\alpha \in (120^\circ, 150^\circ]$, it is considered as a diagonal line. Fig. 4.5 shows some samples of horizontal, diagonal and vertical lines.

The proposed extended set of uniform patterns consist of 48 patterns, as shown in Fig. 4.6. Including 58 uniform patterns, we derive the extended uniform patterns. Similarly as NRLBP, we can derive the extended NRLBP (ENRLBP). Instead of forming uniform patterns, we form extended uniform patterns as our ENRLBP pattern. In such a way, line patterns are preserved during the encoding process. The number of bins of ENRLBP histogram is 107, which is smaller than LTP histogram that has 118 bins.

Figure 4.4: Local primitives detected by $LBP_{8,2}^{u2}$.

Figure 4.5: Samples of line patterns. Those three rows are corresponding to horizontal, diagonal and vertical lines. The diagonal lines are rare patterns for natural images and hence discarded. The remaining horizontal and vertical lines are the proposed extended set of uniform patterns.

4.4 Experimental Results

We conduct comprehensive experiments to validate the advantages of the proposed NRLBP and ENRLBP. Table 4.1 summarizes the approaches compared with, the classifiers used and the applications tested on. The proposed approaches are compared with uniform LBP and uniform

Figure 4.6: 48 extended uniform patterns of ENRLBP.

Table 4.1: Summary of the approaches compared with, the classifiers used and the application tested on.

The approaches	The classifier	The applications
LBP [137]	Nearest-neighbor classifier + Chi-square distance	Face recognition on the AR database [132]
LTP [1]	Nearest-neighbor classifier + histogram intersection distance	Face recognition on the extended Yale B database [134, 135]
Dominant LBP [29]	Nearest-neighbor classifier + modified G-statistic	Face recognition on the O2FN database [136]
Fuzzy LBP [42, 43, 45]	Linear SVM	Texture Recognition on the Outex-13 dataset [133]
Novel extended LBP [39]	K-means clustering	Protein cellular classification on the 2D Hela database [138]
Noise tolerant LBP [40]		Image segmentation on the Outex segmentation dataset [133]

LTP. E.g. for face recognition, $LBP_{8,2}^{u2}$ and $LTP_{8,2}^{u2}$ are used. Let $NRLBP_{P,R}$, $ENRLBP_{P,R}$ denote the coding schemes for NRLBP and ENRLBP using P neighbors at the distance of R to the center pixel, respectively. The number of features for each patch is 59 for $LBP_{8,2}^{u2}$, 118 for $LTP_{8,2}^{u2}$, 59 for $NRLBP_{8,2}$ and 107 for $ENRLBP_{8,2}$. Dominant LBP (DLBP) [29], novel extended LBP (NELBP) [39] and noise tolerant LBP (NTLBP) [40] are compared as they extract information from non-uniform bins, similarly as our approaches do. We choose the dominant patterns that account for 80% of the total pattern occurrences, same as in [29]. Fuzzy LBP (FLBP) [42, 43, 45] is also compared. We implement fuzzy LBP using piece-wise linear fuzzy membership function in [42]:

$$f_{1,d}(z_p) = \begin{cases} 0 & \text{if } z_p < -d, \\ 0.5 + \frac{0.5z_p}{d} & \text{if } -d \leq z_p \leq d, \\ 1 & \text{if } z_p > d. \end{cases} \quad (4.8)$$

$$f_{0,d}(z_p) = 1 - f_{1,d}(z_p). \quad (4.9)$$

where $f_{1,d}$ and $f_{0,d}$ are the probability that pixel difference z_p should be encoded as 1 and 0, respectively. The parameter d controls the amount of fuzzification.

Different classifiers are utilized in our experiments. For face recognition, we use the nearest-neighbor (NN) classifier with three different distance measures: Chi-square distance, histogram intersection distance and G-statistic, as defined in Eqn. (4.10), (4.11) and (4.13), respectively. For texture recognition and protein cellular classification, linear SVM is used, and for image segmentation, k-means clustering algorithms is used.

$$\chi^2(\mathbf{x}, \mathbf{y}) = \sum_{i,j} \frac{(x_{i,j} - y_{i,j})^2}{x_{i,j} + y_{i,j}}, \quad (4.10)$$

$$\mathbf{D}_{HI}(\mathbf{x}, \mathbf{y}) = - \sum_{i,j} \min(x_{i,j}, y_{i,j}), \quad (4.11)$$

$$\mathbf{D}_G(\mathbf{x}, \mathbf{y}) = - \sum_{i,j} x_{i,j} \log y_{i,j}, \quad (4.12)$$

where $\mathbf{x}, \mathbf{y}$ are the concatenated LBP feature vectors of two image samples; $x_{i,j}$ and $y_{i,j}$ are j -th dimension of i -th patch. The G-statistic is numerically unstable, as many histogram bins may have zero elements, which easily causes $D_G \rightarrow \text{inf}$. Thus, we modify it into a numerically stable form:

$$\mathbf{D}_G(\mathbf{x}, \mathbf{y}) = - \sum_{i,j} x_{i,j} \log(x_{i,j} + y_{i,j}), \quad (4.13)$$

Only when both $x_{i,j}$ and $y_{i,j}$ are zero, we set $0 \log(0) = 0$. We call this distance measure as Modified G-statistic (MG). MG is numerically more stable and hence can better handle the

problem of too few elements in the histogram than G-statistic.

We conduct comparison experiments for various applications. Firstly, we inject Gaussian noise and uniform noise of various noise levels onto the images of the AR database [132] for face recognition and the Outex-13 dataset [133] for texture recognition. The proposed NRLBP and ENRLBP are compared with various LBP/LTP variants in order to validate the noise-resistance property of the proposed approaches. Then, we apply the proposed approaches on real images that are noise-prone. Illumination variation is one of big challenges for face recognition. We conduct experiments on two challenge face databases with large illumination variations: the extended Yale B database [134, 135] and the O2FN database [136]. The proposed approaches are also compared with LBP/LTP variants for protein cellular classification on the 2D Hela database [138] and image segmentation on the image of the Outex segmentation database [133] and one image from the web. In order to reduce the illumination variations, the images of the Outex-13 dataset, the extended Yale B database and the O2FN database are pre-processed similarly as in [1]. We utilize the source codes provided by the authors of [1] to perform this photometric normalization.

4.4.1 Face Recognition on the AR Database

For face recognition, we adopt a challenging experimental setting. Only one image per subject is used as the gallery (or training) set and all others are used as the probe set. In many real applications, we are not able to obtain multiple images per subject and we may have only one image per subject.

On the AR database, the proposed approaches are compared with LBP/LTP variants on images injected with noise in order to demonstrate their noise-resistant property. The AR database is of high resolution and high image quality, and considered as a face database with almost no image noise. 75 subjects are chosen from the AR database, each with 14 images. For each

subject, it contains images from 2 sections. Each section contains 7 images: one neutral image, 3 images with different facial expressions and 3 images in different illumination conditions. We repeat experiments 6 times. For each trial, we use Image 1, 5, 6, 8, 12, 13 of each subject as the gallery set, respectively. The other 13 images of each subject are used as the probe set. It is a challenging experimental setting as face images with facial expression variations need to be identified just based on a single face image.

4.4.1.1 Resistant to Additive Gaussian Noise

Gaussian noise is one of the most common types of noise. The images are normalized in the range of $(0, 1)$, and then we apply additive Gaussian noise with zero mean and standard derivation of σ . We conduct the experiments for $\sigma = 0.05, 0.10, 0.15$. The samples of noisy images are shown in Fig. 4.7. When the noise level is high, the images are barely recognizable, and the recognition task becomes more challenging.

For LTP, NRLBP and ENRLBP, there is one free parameter: threshold $t \in [0, 255]$. Fuzzy LBP also has a free parameter: fuzzification d . We vary t for LTP, NRLBP and ENRLBP, and d for fuzzy LBP. Only the recognition rates at the optimal setting are reported. Table 4.2 summarizes the average recognition rate and the standard derivation of each approach at the optimal setting on the AR database injected with Gaussian noise. Table 4.2 shows that the proposed NRLBP and ENRLBP achieve comparable or slightly better performance compared with

Figure 4.7: The images with additive Gaussian noise of $\sigma = 0, 0.05, 0.1, 0.15$, respectively.

Table 4.2: Summary of the average recognition rate, the standard derivation, the optimal threshold t for LTP, NRLBP, ENRLBP and the optimal d for FLBP on the AR database injected with Gaussian noise.

Algorithm	Chi-square Distance, $\sigma =$			Histogram Intersection, $\sigma =$			Modified G-Statistics, $\sigma =$		
	0.05	0.10	0.15	0.05	0.10	0.15	0.05	0.10	0.15
LBP	83.44% $\pm 1.44\%$	64.65% $\pm 2.92\%$	40.91% $\pm 6.52\%$	81.64% $\pm 1.58\%$	55.18% $\pm 7.37\%$	34.34% $\pm 4.38\%$	79.04% $\pm 1.71\%$	56.63% $\pm 3.02\%$	34.56% $\pm 5.93\%$
LTP	83.98% $\pm 1.40\%$ (10)	65.09% $\pm 5.88\%$ (24)	43.78% $\pm 9.96\%$ (8)	81.85% $\pm 2.08\%$ (4)	55.26% $\pm 12.07\%$ (8)	37.69% $\pm 10.77\%$ (8)	80.26% $\pm 0.90\%$ (3)	62.58% $\pm 2.44\%$ (10)	42.74% $\pm 3.74\%$ (24)
DLBP	85.11% $\pm 0.83\%$	62.82% $\pm 6.92\%$	39.57% $\pm 9.76\%$	85.47% $\pm 1.74\%$	62.44% $\pm 7.09\%$	39.03% $\pm 9.87\%$	84.24% $\pm 1.68\%$	61.26% $\pm 5.46\%$	33.49% $\pm 9.04\%$
FLBP	83.95% $\pm 1.32\%$ (12)	78.19% $\pm 1.08\%$ (22)	71.04% $\pm 1.70\%$ (32)	84.17% $\pm 1.42\%$ (18)	74.26% $\pm 2.01\%$ (18)	59.86% $\pm 3.06\%$ (24)	81.44% $\pm 1.41\%$ (8)	74.97% $\pm 1.45\%$ (24)	68.77% $\pm 2.14\%$ (32)
NELBP	65.50% $\pm 3.19\%$	34.82% $\pm 3.01\%$	18.32% $\pm 1.35\%$	64.46% $\pm 3.84\%$	31.56% $\pm 1.81\%$	16.32% $\pm 1.07\%$	66.51% $\pm 3.27\%$	34.85% $\pm 2.23\%$	17.85% $\pm 1.69\%$
NTLBP	63.71% $\pm 3.32\%$	28.94% $\pm 1.53\%$	13.42% $\pm 1.48\%$	67.25% $\pm 3.37\%$	35.69% $\pm 2.85\%$	17.83% $\pm 1.44\%$	61.01% $\pm 3.40\%$	25.35% $\pm 1.62\%$	11.47% $\pm 1.64\%$
NRLBP	85.33% $\pm 1.43\%$ (6)	79.93% $\pm 0.79\%$ (10)	70.67% $\pm 2.90\%$ (18)	85.88% $\pm 0.96\%$ (6)	78.65% $\pm 1.22\%$ (12)	67.62% $\pm 4.87\%$ (28)	84.92% $\pm 1.29\%$ (6)	79.08% $\pm 1.03\%$ (12)	70.55% $\pm 3.22\%$ (20)
ENRLBP	85.98% $\pm 1.35\%$ (3)	80.43% $\pm 0.97\%$ (10)	71.71% $\pm 1.79\%$ (16)	86.02% $\pm 1.09\%$ (6)	80.24% $\pm 1.55\%$ (12)	68.77% $\pm 3.05\%$ (16)	85.42% $\pm 1.03\%$ (6)	80.58% $\pm 0.72\%$ (10)	72.43% $\pm 1.50\%$ (18)

FLBP, whereas consistently outperform other approaches for all settings using different distance measures. As the noise level increases, the performance gain of the proposed approaches over approaches other than FLBP becomes more significant.

In order to study the effect of threshold t (or fuzzification parameter d), we plot the recognition rates vs. t (or d) for LTP, FLBP, NRLBP, ENRLBP using Chi-square distance, as shown in Fig. 4.8. LBP and DLBP are shown as dashed lines. For the low noise level, $\sigma = 0.05$, NRLBP and ENRLBP are slightly better than DLBP and visibly better than LBP, LTP and FLBP. For the middle noise level, $\sigma = 0.10$, the two proposed approaches slightly outperform FLBP and significantly outperform LBP, LTP and BLBP. For the high noise level, $\sigma = 0.15$, while LBP, LTP and DLBP fail to work, FLBP, NRLBP and ENRLBP can still achieve recognition rates

Figure 4.8: The recognition rates of LBP, LTP, DLBP, FLBP, NRLBP and ENRLBP using Chi-square distance vs. threshold t on the AR database injected with Gaussian noise $\sigma = 0.05, 0.10, 0.15$. As the noise level increases, the optimal threshold increases.

over 70% if proper thresholds are applied. Fig. 4.8 shows that the two proposed approaches and FLBP are the only ones that work well for all tested noise levels.

We can also observe from Fig. 4.8 that the optimal threshold increases when the noise level increases. The gradual change of face image carries important information, and will result in small pixel differences. A small threshold will be sufficient to handle the small image noise. If the threshold becomes larger, more pixel differences will be wrongly encoded as *uncertain* state, and the performance will drop as shown in Fig. 4.8(a). When the noise level is high, the pixel differences spread out and the histogram becomes flat. A large threshold is needed to

handle the large image noise.

4.4.1.2 Resistant to Additive Uniform Noise

Uniform noise is another common type of noise. We conduct experiments on the AR database injected with additive uniform noise in the range of $(-p/2, p/2)$. The corresponding standard derivation is $\sigma_u = p/\sqrt{12}$. We vary the noise range for $p = 0.1, 0.2, 0.4, 0.7$, and respectively $\sigma_u = 0.0289, 0.0577, 0.1155, 0.2021$. Sample images are shown in Fig. 4.9. When the noise level is high, the images are severely distorted and barely recognizable.

The proposed approaches are compared with 6 LBP/LTP variants on the AR database in-

Figure 4.9: The images with uniform noise of $p = 0.1, 0.2, 0.4, 0.7$, respectively.

Table 4.3: Summary of the average recognition rate and the standard derivation of each approach at the optimal setting on the AR database injected with uniform noise.

Algorithm	Chi-square Distance, $p =$				Histogram Intersection, $p =$				Modified G-Statistics, $p =$			
	0.1	0.2	0.4	0.7	0.1	0.2	0.4	0.7	0.1	0.2	0.4	0.7
LBP	87.57% $\pm 1.37\%$	81.74% $\pm 2.29\%$	53.81% $\pm 9.01\%$	26.62% $\pm 5.47\%$	87.13% $\pm 1.13\%$	78.46% $\pm 1.88\%$	45.85% $\pm 11.55\%$	25.33% $\pm 5.51\%$	84.94% $\pm 1.30\%$	77.18% $\pm 3.05\%$	46.24% $\pm 7.50\%$	21.66% $\pm 2.56\%$
LTP	87.83% $\pm 1.07\%$	82.80% $\pm 1.42\%$	62.27% $\pm 4.52\%$	32.55% $\pm 3.82\%$	87.50% $\pm 1.13\%$	79.15% $\pm 1.54\%$	48.41% $\pm 12.91\%$	26.77% $\pm 5.98\%$	85.20% $\pm 1.27\%$	77.64% $\pm 3.43\%$	54.56% $\pm 4.60\%$	27.01% $\pm 4.10\%$
DLBP	88.44% $\pm 1.10\%$	83.25% $\pm 1.26\%$	54.44% $\pm 11.78\%$	24.79% $\pm 6.28\%$	89.59% $\pm 1.28\%$	82.72% $\pm 1.78\%$	51.20% $\pm 13.54\%$	24.39% $\pm 8.19\%$	88.48% $\pm 1.28\%$	81.42% $\pm 1.37\%$	46.29% $\pm 13.58\%$	8.92% $\pm 1.61\%$
FLBP	87.23% $\pm 1.33\%$	82.87% $\pm 1.26\%$	75.26% $\pm 2.04\%$	50.39% $\pm 3.23\%$	87.11% $\pm 1.22\%$	82.65% $\pm 1.29\%$	68.53% $\pm 3.07\%$	41.47% $\pm 2.11\%$	85.25% $\pm 1.31\%$	80.55% $\pm 1.71\%$	72.31% $\pm 2.16\%$	46.63% $\pm 4.04\%$
NELBP	74.77% $\pm 5.06\%$	60.00% $\pm 5.33\%$	24.60% $\pm 4.29\%$	15.52% $\pm 4.26\%$	75.73% $\pm 4.18\%$	58.14% $\pm 4.66\%$	23.25% $\pm 3.60\%$	15.83% $\pm 3.55\%$	75.79% $\pm 4.38\%$	60.56% $\pm 4.57\%$	24.19% $\pm 4.08\%$	15.16% $\pm 3.61\%$
NTLBP	73.88% $\pm 5.70\%$	57.35% $\pm 3.19\%$	20.80% $\pm 1.08\%$	8.67% $\pm 1.29\%$	75.71% $\pm 4.85\%$	62.39% $\pm 2.77\%$	25.01% $\pm 2.38\%$	11.09% $\pm 1.67\%$	73.88% $\pm 5.23\%$	54.07% $\pm 4.04\%$	16.82% $\pm 1.61\%$	7.61% $\pm 1.26\%$
NRLBP	88.07% $\pm 0.89\%$	84.48% $\pm 1.09\%$	77.26% $\pm 2.55\%$	56.07% $\pm 5.15\%$	88.79% $\pm 0.81\%$	85.06% $\pm 1.10\%$	75.76% $\pm 2.94\%$	53.61% $\pm 6.95\%$	87.38% $\pm 1.25\%$	83.90% $\pm 0.80\%$	76.38% $\pm 2.04\%$	55.90% $\pm 4.94\%$
ENRLBP	88.68% $\pm 0.91\%$	84.51% $\pm 1.70\%$	77.37% $\pm 1.46\%$	56.22% $\pm 3.51\%$	88.89% $\pm 0.89\%$	84.97% $\pm 1.49\%$	76.09% $\pm 2.16\%$	51.74% $\pm 5.29\%$	87.81% $\pm 1.15\%$	84.43% $\pm 1.02\%$	76.99% $\pm 1.05\%$	55.76% $\pm 4.41\%$

jected with uniform noise. The average recognition rates and the standard derivation at the optimal setting are summarized in Table 4.3. Both proposed approaches achieve comparable or better performance than other approaches. DLBP performs well for very low noise level, but it is even more sensitive to noise than LBP and hence performs even worse than LBP for middle and high noise levels. FLBP is also shown resistant to noise. Except for FLBP, as the noise level increases, the performance gain of the proposed approaches over other approaches increases.

4.4.2 Texture Recognition on Outex-13 dataset

Outex-13 dataset [133] consists of 68 classes of textures, each with 20 images. To test the noise-resistant property of the proposed approaches on the applications other than face recognition, we inject Gaussian noise and uniform noise of different noise levels onto the images of Outex-13 dataset, e.g. Gaussian noise of $\sigma = 0.05, 0.10, 0.15$ and uniform noise of $p = 0.1, 0.2, 0.4$. Preprocessing in [1] is useful to reduce noise. Thus, the noisy images are preprocessed in the same way as in [1]. Sample images and preprocessed images are shown in the first and second row of Fig. 4.10, respectively.

Figure 4.10: Row 1 shows the sample images of Outex-13 dataset injected with Gaussian noise of $\sigma = 0.05, 0.10, 0.15$ and uniform noise of $p = 0.1, 0.2, 0.4$, respectively. Row 2 shows the respective images after the preprocessing as in [1].

Table 4.4: Texture recognition on the Outex-13 database injected with Gaussian noise and uniform noise.

Algorithm	Gaussian Noise			Uniform Noise		
	$\sigma = 0.05$	$\sigma = 0.1$	$\sigma = 0.15$	$p = 0.1$	$p = 0.2$	$p = 0.4$
LBP	52.09% $\pm 1.23\%$	41.71% $\pm 0.96\%$	33.53% $\pm 0.34\%$	65.21% $\pm 1.01\%$	47.18% $\pm 0.93\%$	38.26% $\pm 0.99\%$
LTP	59.26% $\pm 1.01\%$	50.85% $\pm 1.16\%$	43.03% $\pm 1.27\%$	69.74% $\pm 1.07\%$	54.50% $\pm 1.95\%$	51.74% $\pm 2.10\%$
DLBP	52.47% $\pm 1.02\%$	41.26% $\pm 1.53\%$	32.74% $\pm 1.38\%$	64.62% $\pm 0.74\%$	47.00% $\pm 1.20\%$	38.29% $\pm 1.20\%$
FLBP	61.29% $\pm 2.59\%$	53.32% $\pm 1.58\%$	42.85% $\pm 1.49\%$	69.56% $\pm 0.96\%$	55.65% $\pm 1.32\%$	52.53% $\pm 1.55\%$
NELBP	34.50% $\pm 1.83\%$	33.59% $\pm 1.68\%$	26.76% $\pm 0.68\%$	42.59% $\pm 1.32\%$	29.76% $\pm 1.46\%$	30.88% $\pm 2.34\%$
NTLBP	32.09% $\pm 2.26\%$	31.56% $\pm 0.69\%$	26.68% $\pm 1.19\%$	40.91% $\pm 1.39\%$	27.00% $\pm 0.76\%$	29.24% $\pm 1.93\%$
NRLBP	62.88% $\pm 0.95\%$	54.62% $\pm 0.74\%$	43.24% $\pm 1.00\%$	71.50% $\pm 1.27\%$	58.15% $\pm 1.09\%$	55.41% $\pm 1.78\%$
ENRLBP	63.09% $\pm 1.96\%$	55.06% $\pm 1.66\%$	44.26% $\pm 1.00\%$	70.94% $\pm 1.15\%$	58.09% $\pm 0.48\%$	54.76% $\pm 0.58\%$

We randomly choose 10 images from each class for training and the rest for testing. The proposed approaches are compared with 6 LBP/LTP variants. We extract features using 8 neighbors at the radius of one. Linear SVM is used as the classifier, which is implemented using LIBSVM package [139]. The cost parameter C is chosen as 1. The experiments are repeated 5 times, and only the average performance is reported. Table 4.4 summarizes the performance comparison on the Outex-13 dataset injected with Gaussian noise and uniform noise. The proposed NRLBP and ENRLBP consistently achieve comparable or better performance compared with other approaches.

4.4.3 Face Recognition on the Extended Yale B Database

The extended Yale B database [134, 135] contains 38 subjects under 9 poses and 64 illumination conditions. We follow the same database partition as in [1]. The images with most neutral

Figure 4.11: The 1st row and 2nd row show the samples of geometrically normalized and photometrically normalized images for the extended Yale B database, respectively. The leftmost image is the gallery image, and the other 3 images taken under extreme lighting conditions are the probe images.

light source (“A+000E+00”) are used as the gallery images and all other frontal images are used as the probe images (in total 2414 images of 38 subjects). This dataset contains large illumination variations. The sample images are shown in the first row of Fig. 4.11. Some images are taken under extreme lighting conditions. Even after photometric normalization, as shown in the second row of Fig. 4.11, a large amount of image noise exist in the images. The proposed approaches are compared with 6 LBP/LTP variants using nearest-neighbor classifier with Chi-square distance, histogram intersection and modified G-statistic. Table 4.5 summarizes the highest recognition rates at the optimal threshold for various approaches using different distance measures. The proposed approaches achieve a slightly better performance than LBP, LTP, DLBP and FLBP, and much better performance than NELBP and NTLBP.

4.4.4 Face Recognition on the O2FN Mobile Database

The O2FN mobile face database [136] is our in-house face database. It is designed to evaluate the face recognition algorithms on mobile face images, which are of low resolution and low image quality, and significantly corrupted by the noise. It contains 2000 face images of size 144×176 pixels from 50 subjects. The images are self-taken by the users. The users are told to

Table 4.5: The face recognition rate and the optimal threshold on the extended Yale B database.

Algorithm	Chi-square Distance	Histogram Intersection	Modified G-Statistics
LBP	96.07%	93.32%	96.12%
LTP	98.25% (10)	97.99% (10)	98.29% (8)
DLBP	96.12%	97.83%	98.45%
FLBP	98.45% (9)	98.16% (9)	98.54% (12)
NELBP	81.91%	82.29%	84.92%
NTLBP	80.37%	80.16%	83.12%
NRLBP	98.71% (9)	98.66% (8)	98.66% (11)
ENRLBP	98.75% (9)	98.66% (8)	98.62% (10)

take roughly 20 indoor images and 20 outdoor images with minimum facial expression variations and out-plane rotations. Thus, the O2FN database mainly contains in-plane rotations and illumination variations. Fig. 4.12 shows some samples of geometrically normalized and photometrically normalized images. The images are captured by O2 XDA frontal camera with native phone settings and without post-processing. The images are severely distorted by the noise, e.g. Gaussian noise, Salt & Pepper noise and motion blur. To reduce the noise and illumination variations, the images are photometric normalized as in [1]. Even after the photometric normalization, as shown in Fig. 4.12, the images still contain a large amount of noise.

The proposed approaches are compared with 6 LBP/LTP variants using nearest-neighbor

Figure 4.12: The samples of geometrically normalized (Row 1) and photometrically normalized (Row 2) face images of the O2FN databases.

classifier with 3 different distance measures. The experiments are repeated 5 times. For each trial, we randomly choose one image of each subject as the gallery set and the rest as the probe set. We test LTP, NRLBP and ENRLBP for different thresholds, and FLBP for different d . Only the performance at the optimal setting is reported. The average recognition rates and the standard derivation at the optimal setting on the O2FN database are summarized in Table 4.6. The proposed NRLBP and ENRLBP achieve a comparable or slightly better performance compared with LTP, DLBP and FLBP, and significantly outperform LBP, NELBP and NTLBP using all three distance measures.

Table 4.6: Performance comparison for face recognition on the O2FN database.

Algorithm	Chi-square Distance	Histogram Intersection	Modified G-Statistics
LBP	76.59% $\pm$ 1.60%	74.14% $\pm$ 1.44%	75.18% $\pm$ 1.15%
LTP	78.88% $\pm$ 1.65%	76.88% $\pm$ 1.91%	78.16% $\pm$ 1.39%
DLBP	78.07% $\pm$ 1.69%	79.88% $\pm$ 2.10%	79.01% $\pm$ 2.04%
FLBP	80.24% $\pm$ 1.58%	79.15% $\pm$ 1.49%	80.01% $\pm$ 1.46%
NELBP	56.74% $\pm$ 1.75%	58.25% $\pm$ 2.16%	59.12% $\pm$ 1.73%
NTLBP	56.96% $\pm$ 1.82%	58.40% $\pm$ 1.65%	58.63% $\pm$ 2.18%
NRLBP	80.76% $\pm$ 1.56%	80.29% $\pm$ 1.63%	80.68% $\pm$ 1.57%
ENRLBP	81.66% $\pm$ 1.83%	81.28% $\pm$ 1.80%	81.44% $\pm$ 1.91%

4.4.5 Protein Cellular Classification on 2D Hela Database

Protein cellular classification is useful when characterizing newly discovered genes. 2D Hela database contains 862 single-cell images (16-bit gray scale of size 382×382 pixels) [138]. There are ten classes in this database and each with more than 70 images. Some sample images are shown in Fig. 7.9.

Multi-scale LBP has shown good performance on this dataset [140]. We use $\{P, R\}$ to represent the descriptor extracted using P neighbors at the distance of R to the center pixel. Then, we extract features at multiple scales: $\{8, 1\}$, $\{8, 2\}$ and $\{8, 3\}$. Then those features are concatenated as the final feature vector for classification. Linear SVM [139] is used for

Figure 4.13: Sample images of the 2D Hela database.

classification. The cost parameter is the same as in [41], i.e. $C = 100$. We randomly choose 80% of the database for training and 20% for testing. The experiments are repeated five times and the average performance is reported. The performance comparison of the proposed approaches with other LBP/LTP variants are shown in Table 4.7. The proposed NRLBP achieves the highest recognition rate of 95.93%. FLBP, LTP and the proposed ENRLBP also achieve good performance on this dataset.

Table 4.7: The performance comparison for protein cellular classification on the 2D Hela database in terms of recognition rate and time.

Algorithm	Average recognition rate	Time (ms)
LBP	89.53% $\pm$ 0.41%	64.7
LTP	95.70% $\pm$ 1.91%	95.5
DLBP	88.14% $\pm$ 1.95%	109.2
FLBP	93.26% $\pm$ 1.87%	2823.7
NELBP	91.16% $\pm$ 1.04%	109.4
NTLBP	88.60% $\pm$ 1.57%	110.3
NRLBP	95.93% $\pm$ 1.48%	87.5
ENRLBP	95.58% $\pm$ 0.66%	103.6

4.4.6 Image Segmentation

Besides recognition tasks, we also conduct comparison experiments for image segmentation. We extract features using 8 neighbors at the distance of one to the center. We follow the similar setup in [45] to segment texture images. Features are extracted in a raster-scanning way using sliding windows of 16×16 pixels with a step size of one pixel. K-means clustering algorithm with “cityblock” distance is used to classify the scanning windows. The number of clusters is given as input. Qualitative segmentation results of the proposed approaches and LBP/LTP variants are given. We choose the tenth image of the Outex segmentation test suite [133] and one natural scene image downloaded from the internet for illustration, as shown in Fig. 4.14 and Fig. 4.15.

The Outex image consists of five textures. The ground-truth labeling of segments is shown in Fig. 4.14 (b). Apparently, the proposed NRLBP and ENRLBP achieve better segmentation

Figure 4.14: Example of segmentation results on a mix-texture image for LBP, LTP, dominant LBP, fuzzy LBP, the proposed NRLBP and ENRLBP.

Figure 4.15: Example of segmentation results on a natural scene image for LBP, LTP, dominant LBP, fuzzy LBP, the proposed NRLBP and ENRLBP.

performance than LBP, LTP, DLBP and FLBP. The natural scene image shown in Fig. 4.15(a) has four texture regions: sky, far forest, nearby forest and grassland. The proposed NRLBP and ENRLBP have much less misclassifications, and hence achieve better performance than other approaches.

4.4.7 Comparison of Computational Complexity

The proposed NRLBP and ENRLBP can be implemented by a look-up table to compute the NRLBP/ENRLBP histogram from the uncertain code. It is very fast to compute the contribution of an *uncertain* code to the histogram by the look-up table and hence derive the feature vector of NRLBP/ENRLBP during recognition. The average time per image of feature extraction on the 2D Hela database for various LBP/LTP variants is shown in Table 4.7. The image is of size 382×382 pixels. Features are extracted under the setting of $P = 8, R = 1$. We use Matlab 2012b on Intel Duo CPU 3.0 GHz with 4Gb RAM. Compared with LBP, NRLBP and ENRLBP only introduce a small overhead. NRLBP is in fact the second fastest approach. In

contrast, it takes much more time to compute FLBP features, e.g. 2823.7 ms, which is 32 times of NRLBP. Due to too many nested loops required, it is very time-consuming to compute the FLBP histogram using Matlab implementation.

4.5 Summary

LBP is sensitive to noise. Even a small noise may change the LBP code significantly. LTP partially solves this problem by encoding the small pixel differences into a separate state. However, both LBP and LTP lack a mechanism to recover the corrupted image local structures.

As the small pixel difference is most vulnerable to noise, we encode it as *uncertain* bit first, and then determine its value based on the other bits of the LBP code to form a code of image local structure. Uniform patterns represent local image primitives, and appear more frequently than non-uniform patterns in natural images. In contrast, non-uniform patterns are less reliable, thus are more error-prone. Therefore, we assign the values of *uncertain* bits so as to form all possible uniform LBP codes. In such a way, we correct noisy non-uniform patterns back to uniform code. For LBP and LTP, a large group of local primitives, i.e. line patterns, are completely ignored. Thus, we propose extended uniform patterns and form those patterns as our ENRLBP patterns when determine *uncertain* bits.

The proposed approaches show stronger noise-resistance compared with other approaches. We inject Gaussian noise and uniform noise of different noise levels on the AR database for face recognition and the Outex-13 dataset for texture recognition. Compared with FLBP, the proposed approaches are much faster and achieve comparable or slightly better performance. They consistently achieve better performance than all other approaches. We further compare the proposed NRLBP and ENRLBP with others for face recognition on the extended Yale B database and the O2FN database, protein cellular classification on the 2D Hela database, as well as image segmentation. The proposed approaches demonstrate a superior performance on

these applications.

Chapter 5

Learning LBP Structure by Maximizing the Conditional Mutual Information

LBP features of more bits extracted in a large structure can improve the recognition performance, but at the cost of high feature dimensionality. Thus, the large LBP structure is often broken into small handcrafted ones. The handcrafted LBP structure may not be optimal as it is often selected heuristically. In this chapter, we propose a new framework to search for the optimal LBP structures for a specific application¹. Our objective is to select an optimal subset of binarized pixel differences to compose the LBP structure. In view of the strong correlation of binarized pixel difference, we propose an incremental maximal-conditional-mutual-information scheme for LBP structure learning. The proposed approach demonstrates a superior performance on texture classification, scene recognition and dynamic texture recognition.

5.1 Introduction

One potential problem surfaces with the wide application of LBP features, *i.e.* the feature dimensionality increases exponentially with the number of LBP bits. The histogram of original

¹This work is partially published in International Conference on Image Processing 2013 (conference paper [6] in author's publication list), IEEE Signal Processing Letters (journal paper [3] in author's publication list), and Pattern Recognition (journal paper [4] in author's publication list).

LBP has 256 bins only [137]. This LBP only utilizes the pixel differences between a pixel and its 8 nearest neighbors, which cannot capture the image structures of a larger scale. In [16], LBP features were extracted using P neighbors uniformly sampled on a circle at the radius of R to the center pixel, denoted as $LBP_{P,R}$. By varying R , micro-structures at different scales are captured. $LBP_{P,R}$ has 2^P bins. Due to the storage and computational complexity constraints, the number of LBP bits is in general limited to 24, *i.e.* $2^{24} = 16777216$ bins. Instead of using circular neighbors, some other geometries were explored in Local Quantized Pattern (LQP) [2], *e.g.* horizontal line, vertical line, horizontal-vertical cross, diagonal cross and disc shape. Recent researches [2, 16, 18] show that LBP features using more bits extracted in a larger neighborhood have higher potential to capture complex patterns than using the basic ones, at the cost of high feature dimensionality. It imposes a big challenge to handle such high-dimensional LBP features, especially when we need to extract these features from a large-scale database.

Instead of constructing a high-dimensional LBP histogram, some algorithms tackle the problem by breaking the large structure into small ones, or simply replacing it by a small one. LBP-TOP [19], CS-LTP [20] and LQP [2] are initial attempts to reduce the dimensionality by directly reducing the size of LBP structure. A heuristic hill-climbing technique was used to select the LBP structures in [78]. However, these heuristically selected structures may not be optimal. In [141], Lei et al. proposed to iteratively learn discriminant image filters and neighborhood sampling strategy. Their approach works well for face recognition, but cannot effectively handle large image variations in other applications.

In this chapter, we propose to reduce the dimensionality of LBP features by optimizing the LBP structure directly. We formulate it as a point-selection problem. Given a neighborhood, the goal is to select an optimal subset of neighbors to compose the LBP structure. For each point, its binarized pixel difference with respect to the center pixel is treated as a feature. For feature selection, it is often desirable to maximize the dependency of target class on data distribution (known as Max-Dependency scheme). However, it is difficult to directly calculate such a de-

pendency as it requires to estimate high-order multivariate probability density functions [142]. Thus, approximated algorithms were often utilized, e.g. Max-Relevance, Min-Redundancy-Max-Relevance [34, 142] and Maximal-Joint-Mutual-Information [143]. One important characteristic of the LBP-structure-learning problem is that these binarized-pixel-difference features are strongly correlated. However, previous approaches assume certain degree of feature independence, e.g. the high-order interaction is assumed negligible in [34, 143]. In view of this, we propose to first approximate the dependency as closely as possible by a set of low-order conditional mutual information, and then achieve Max-Dependency criterion through maximizing its approximation. In such a way, we derive a more accurate approximation of Max-Dependency criterion. Then, we propose a Maximal-Conditional-Mutual-Information (MCMI) scheme for LBP structure learning. It is difficult to seek a globally optimal solution for MCMI scheme. For simplicity, we utilize sequential forward selection (SFS) [144].

After deriving the LBP structures, the LBP histograms are generated using these structures. PCA is applied on the histogram of each patch to further reduce the dimensionality. The final concatenated feature vector is classified by a support vector machine with a RBF kernel or a nearest-neighbor classifier with χ^2 distance. We use LIBSVM package [139] in the implementation.

5.2 Related Work

We review the related LBP variants in this section only. The review of feature-selection approaches is given Section 5.3.1. Various approaches have been proposed to handle high dimensionality of LBP features. In uniform LBP [16], all non-uniform patterns are grouped into one bin, and hence the feature dimensionality is reduced from 2^P to $P(P - 1) + 3$. CENTRIST feature [3] utilizes a handcrafted 3×3 neighborhood. By applying principal component analysis on the LBP histogram, it reduces the feature dimensionality of each patch from 256 to 40. In

LQP [2], many handcrafted LBP structures are built from a large neighborhood. The feature dimensionality is reduced by clustering the histogram bins. The spatial-temporal LBP has a typical large structure. In LBP-TOP [19], the large structure is broken into three small ones, *i.e.* the LBP histograms are extracted from three orthogonal planes (XY-plane, XT-plane and YT-plane).

5.3 Overview of Proposed Approach

In general, it is beneficial to extract LBP features using more binarized pixel differences in a larger neighborhood. As the feature dimensionality grows exponentially, those approaches that directly extract/select histogram-bin features quickly become infeasible. An optimal LBP structure needs to be determined first to produce a histogram of a reasonable size. Some initial attempts utilize handcrafted structures [2, 19, 20]. To develop a good handcrafted structure, a trial-and-error process is often involved. Even so, in many situations the handcrafted structure cannot yield a desirable result due to the following: a) Optimality cannot be guaranteed as the handcrafted structure is often selected heuristically. b) The handcrafted structure is not scalable. Given a good handcrafted structure, it is unclear how to compress it to a compact one that achieves a comparable performance at a higher speed, or how to extend it to a larger one to achieve a better performance. c) The handcrafted structure is not universal. For different applications or even different patches of an image, the intrinsic image characteristics may be different. It is impossible to develop one structure that works well for all. Tremendous effort is needed to design a good handcrafted structure for every application and every patch.

The block diagram of the proposed approach is shown in Fig. 5.1. We derive the LBP structure first, and then use it to generate LBP-histogram bins. Formally, denote $z_i = C_i - I_c$ as the pixel difference between the neighboring pixel C_i and the center pixel I_c . The binarized

pixel difference is defined as:

$$x_i = \begin{cases} 1 & \text{if } z_i \geq 0, \\ 0 & \text{if } z_i < 0. \end{cases} \quad (5.1)$$

Now point C_i is represented by its binarized pixel difference x_i . We cast the LBP structure learning as a point subset selection problem: given binarized pixel differences $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ and target classification variable c , the goal is to find a subset of m binarized pixel differences $\mathbf{x}_m \subseteq \mathbf{x}$ that “optimally” characterize c . We solve the problem via the proposed Maximum-Conditional-Mutual-Information scheme, which will be elaborated in detail in the next section. As a result, the feature dimensionality is reduced from 2^n to 2^m . Then, we apply PCA to further reduce it to d .

Potentially, the proposed approach could handle a large set of binarized pixel differences, whereas approaches that directly extract features from the histogram bins [2, 3, 21, 77, 83, 145, 146] cannot as it is not feasible to enumerate 2^n bins for a large n . This is even more crucial

Figure 5.1: Block diagram of the proposed approach. The neighbors of learned LBP structure are marked as “o”.

when handling a large-scale database.

Even in the scenario that it is possible to enumerate all histogram bins, the proposed approach may still yield a better performance. The joint probability $p(\mathbf{x})$ of set $\mathbf{x}$ can be estimated by the LBP histogram of 2^n bins generated from these n points. If point x_j is not selected, the marginal probability of the remaining point set is $\sum_{x_j} p(\mathbf{x})$. Correspondingly, each resulting bin is the summation of two original histogram bins. Eventually, when we derive the LBP structure of m points out of these n candidates, we reduce the number of histogram bins from 2^n to 2^m , and each resulting bin is the summation of 2^{n-m} original bins. Thus, the resulting histogram is less noisy, more statistically significant and reliable than the original histogram. In contrast, for those direct-bin-selection approaches [2, 3, 21, 83] the feature set is directly extracted from the histogram of 2^n bins, which may be statistically insignificant and error-prone. Thus, we expect a better generalization performance for the proposed approach.

Now we briefly discuss how to determine the potential candidates. In LQP [2], disc structure in a neighborhood of 5×5 pixels has demonstrated a superior performance over other geometries. Thus, we use the same neighborhood, and the binarized pixel differences between 24 neighbors and the center pixel as potential candidates, as shown in Fig. 5.2(a). The LBP structure of CENTRIST feature [3] consists of 8 neighbors in the neighborhood of 3×3 pixels, as highlighted in yellow in Fig. 5.2(a). For spatial-temporal LBP (STLBP), we consider a spatial-temporal neighborhood of 3×3 pixels in 3 successive frames, resulting in 26 binarized pixel differences, as shown in Fig. 5.2(b). It is not feasible to directly use all 26 neighbors to construct the histogram as $2^{26} = 67,108,864$. Thus, we treat them as potential candidates and aim to find the optimal subset.

Figure 5.2: Potential candidates. (a) The binarized pixel differences between 24 neighbors and center pixel in a neighborhood of 5×5 pixels as potential candidates. LQP $Disc_5^{3*}$ [2] utilizes the same set of neighbors. CENTRIST feature [3] utilizes 8 neighbors highlighted in yellow. (b) The binarized pixel differences between 26 neighbors and center pixel of Frame t as potential candidates for STLBP.

5.4 An Incremental Maximal-Conditional-Mutual-Information Scheme for LBP Structure Learning

Due to the spatial dependence among image pixels within a small neighborhood, the binarized pixel differences are strongly correlated. Previous feature-selection approaches assume certain degree of feature independence, and hence may not be suitable in our scenario. Thus, we propose an incremental Maximal-Conditional-Mutual-Information scheme. We begin with problem analysis of previous algorithms.

5.4.1 Review of Feature-Selection Algorithms Based on Mutual Information

In Max-Dependency scheme [142], the dependency of target class c on data distribution is maximized. Mutual information is often employed to characterize the dependency. Given two random variables x and y , their mutual information is defined in terms of probability density functions $p(x)$, $p(y)$ and $p(x, y)$:

$$I(x; y) = \int \int p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dx dy. \quad (5.2)$$

Given a set of features $\mathbf{x}$, the goal is to find a subset of m features $\mathbf{x}_m \subseteq \mathbf{x}$, which jointly have the largest dependency on target class c :

$$\mathbf{x}_m^* = \operatorname{argmax}_{\mathbf{x}_m \subseteq \mathbf{x}} I(\mathbf{x}_m; c). \quad (5.3)$$

$$\begin{aligned} I(\mathbf{x}_m; c) &= \int \int p(\mathbf{x}_m, c) \log \frac{p(\mathbf{x}_m, c)}{p(\mathbf{x}_m)p(c)} d\mathbf{x}_m dc \\ &= \int \dots \int p(x_1, \dots, x_m, c) \log \frac{p(x_1, \dots, x_m, c)}{p(x_1, \dots, x_m)p(c)} \\ &\quad dx_1 \dots dx_m dc. \end{aligned}$$

It is difficult to reliably estimate $p(x_1, \dots, x_m)$ and $p(x_1, \dots, x_m, c)$ due to limited training samples available and the large number of joint states to be estimated. Thus, approximated algorithms were often utilized [4, 142]. In Max-Relevance scheme, $I(\mathbf{x}_m; c)$ is approximated by the mean value of mutual information between individual feature x_i and c :

$$\mathbf{x}_m^* = \operatorname{argmax}_{\mathbf{x}_m} \frac{1}{m} \sum_{x_i \in \mathbf{x}_m} I(x_i; c). \quad (5.4)$$

Max-Relevance has a strong assumption that all features are independent to each other [142], which in general does not hold. The features selected according to Max-Relevance may have rich redundancy. To solve this problem, the criterion of Min-Redundancy was added to select mutually exclusive features.

$$\mathbf{x}_m^* = \operatorname{argmin}_{\mathbf{x}_m} \frac{1}{m^2} \sum_{x_i, x_j \in \mathbf{x}_m} I(x_i; x_j). \quad (5.5)$$

In [142], Min-Redundancy and Max-Relevance (mRMR) were combined:

$$\mathbf{x}_m^* = \operatorname{argmax}_{\mathbf{x}_m} \sum_{x_i \in \mathbf{x}_m} I(x_i; c) - \frac{1}{m} \sum_{x_i, x_j \in \mathbf{x}_m} I(x_i; x_j). \quad (5.6)$$

mRMR has two weak assumptions that high-order interaction information is negligible and the desired features are conditional independent given classification variable c [147]. In many cases, these two assumptions do not hold. Particularly in our case, the binarized-pixel-difference features are strongly correlated, and hence the high-order interaction information among these features is non-negligible.

Both Max-Relevance and mRMR only approximate Max-Dependency criterion intuitively. Recent research [147] shows that when high-order interaction information is negligible, $I(\mathbf{x}_m; c)$ can be approximated by:

$$I(\mathbf{x}_m; c) \approx \sum_{x_i \in \mathbf{x}_m} I(x_i; c) - \sum_{x_i, x_j \in \mathbf{x}_m} I(x_i; x_j) + \sum_{x_i, x_j \in \mathbf{x}_m} I(x_i; x_j | c), \quad (5.7)$$

where $I(x_i; x_j | c)$ is conditional mutual information. For discrete random variables x, y, z , the conditional mutual information is defined as:

$$\begin{aligned} I(x; y | z) &= \mathbb{E}_z \{ I(x; y) | z \} \\ &= \sum_{x, y, z} p(x, y, z) \log \frac{p(z) p(x, y, z)}{p(x, z) p(y, z)}, \end{aligned} \quad (5.8)$$

where $\mathbb{E}_z \{ \cdot \}$ is the expectation on z .

It can be observed that only when Eqn. (5.7) is dominated by its first term, $\sum_{x_i \in \mathbf{x}_m} I(x_i; c)$ defined in Eqn. (5.4) for Max-Relevance is a good approximation of $I(\mathbf{x}_m; c)$. It is equivalent to the feature-independence assumption of Max-Relevance. mRMR defined in Eqn. (5.6) differs from Eqn. (5.7) by a missing term $\sum_{x_i, x_j \in \mathbf{x}_m} I(x_i; x_j | c)$ and a weighting factor for the second term. It has the assumption that the last term in Eqn. (5.7) and high-order interaction information

are negligible. In our scenario, these two assumptions do not hold. A better approximation, Maximal-Joint-Mutual-Information (MJMI) scheme, is given in [143]. However, MJMI scheme also assumes that high-order interaction is negligible.

5.4.2 A Close Approximation to $I(\mathbf{x}_m; c)$

In this chapter, we propose to approximate $I(\mathbf{x}_m; c)$ closely by a set of low-order conditional mutual information. Our formulation is inspired by [148]. Chow and Liu approximated high-order discrete probability distributions $P(\mathbf{x})$ with dependence trees $P^t(\mathbf{x})$ as follows:

$$P^t(\mathbf{x}) = \prod_{i=1}^n P(x_{a_i} | x_{a_{j(i)}}), 0 \leq j(i) < i, \quad (5.9)$$

where $(a_1, a_2, \dots, a_n)$ is an unknown permutation of integers $1, 2, \dots, n$ and $P(x_{a_1} | x_{a_0})$ is by definition equal to $P(x_{a_1})$.

Similarly, we would like to approximate $I(\mathbf{x}_m; c)$ by a set of conditional mutual information of the form $I(x_i; c | x_j)$:

$$I^t(\mathbf{x}_m; c) = \sum_{i=1}^m I(x_{b_i}; c | x_{b_{j(i)}}), 0 \leq j(i) < i, \quad (5.10)$$

where $\{b_1, b_2, \dots, b_m\}$ is an unknown permutation of integers $1, 2, \dots, m$ and $I(x_{b_1}; c | x_{b_0})$ is by definition equal to $I(x_{b_1}; c)$. In fact, such a formulation is feasible. Recall the chain rule for $I(\mathbf{x}_m; c)$:

$$I(\mathbf{x}_m; c) = \sum_{i=1}^m I(x_i; c | x_1, \dots, x_{i-1}). \quad (5.11)$$

For $i \geq 3$, $I(x_i; c | x_1, \dots, x_{i-1})$ is high-order conditional mutual information. Compare Eqn. (5.10) with Eqn. (5.11), we can see that if we could approximate high-order conditional mutual information $I(x_i; c | x_1, \dots, x_{i-1})$ closely by low-order conditional mutual information of the form

$I(x_i; c|x_j)$, we could approximate $I(\mathbf{x}_m; c)$ closely by $I^t(\mathbf{x}_m; c)$ as defined in Eqn. (5.10).

Now the question is: how to derive a close approximation of $I(\mathbf{x}_m; c)$ using $I^t(\mathbf{x}_m; c)$ defined in Eqn. (5.10)? More specifically, which term $I(x_i; c|x_j)$ should be chosen? At a first glance, we may assume that $I(x_i; c|x_j) \geq I(x_i; c|x_1, \dots, x_{i-1})$, $j = 1, 2, \dots, i-1$. Thus, $I(\mathbf{x}_m; c)$ is upper-bounded by $I^t(\mathbf{x}_m; c)$, and we should minimize $I(x_i; c|x_j)$ in order to achieve a tighter upper bound. However, this assumption may not hold always. Let us re-examine it:

$$I(x_i; c|x_1, \dots, x_{i-1}) - I(x_i; c|x_j) = I(x_i; \mathbf{x}_k; c|x_j), \quad (5.12)$$

where $I(x_i; \mathbf{x}_k; c|x_j)$ is conditional interaction information and $\mathbf{x}_k = \{x_k\}$, $k = 1, 2, \dots, i-1$, $k \neq j$. The interaction information is the gain (or loss) in information among a set of variables due to additional knowledge of the other variables, *i.e.* $I(x; y; z) = I(x; y|z) - I(x; y)$. The conditional interaction information can be obtained as: $I(x; y; z|w) = \mathbb{E}_w\{I(x; y; z)|w\}$.

In general, negative interaction information seems much more natural than positive interaction information as it explains typical common-cause structures. However, in our case we are dealing with positive interaction information. Let us start with a simplified case. The interaction information among x_i, x_j, c can be derived as: $I(x_i; x_j; c) = I(x_i, x_j; c) - I(x_i; c) - I(x_j; c)$, where $I(x_i, x_j; c)$ is joint mutual information. One feature alone, *i.e.* each individual binarized-pixel-difference feature, is rather weak and does not have much discriminative power. Thus, both $I(x_i; c)$ and $I(x_j; c)$ are small. If we consider two features together, the classification capability increases significantly, *i.e.* $I(x_i, x_j; c) > I(x_i; c) + I(x_j; c)$. This is a typical example of positive interaction information. We can view it from another aspect. By definition $I(x_i; x_j; c) = I(x_j; c|x_i) - I(x_i; c)$. In our case, $I(x_i; c)$ is very small and approaches 0. Then, $I(x_i; x_j; c) \approx I(x_j; c|x_i) \geq 0$. We calculate $I(x_i; x_j; c)$ for the 21-land dataset [149] and dyn-tex++ dataset [5, 150]. We find that 92.9% and 96.4% are positive interaction information, respectively. Similarly, $I(x_i; c; \mathbf{x}_k|x_j)$ may also very likely be positive.

Therefore, we conclude that in general for binarized pixel differences, $I(x_i; c|x_j) \leq I(x_i; c|x_1, \dots, x_{i-1})$. Thus, rather than an upper-bound, $I^t(\mathbf{x}_m; c)$ is actually a lower bound for $I(\mathbf{x}_m; c)$. We will further verify this in the experimental section. In order to obtain a tighter lower bound, $I^t(\mathbf{x}_m; c)$ is derived as:

$$I^t(\mathbf{x}_m; c) = \sum_{i=1}^m \max_{j(i)} I(x_{b_i}; c|x_{b_{j(i)}}), 0 \leq j(i) < i. \quad (5.13)$$

5.4.3 Proposed Incremental MCMI Scheme

After we approximate $I(\mathbf{x}_m; c)$ closely by $I^t(\mathbf{x}_m; c)$ defined in Eqn. (5.13), we achieve Max-Dependency criterion by maximizing $I^t(\mathbf{x}_m; c)$:

$$\mathbf{x}_m^* = \operatorname{argmax}_{\mathbf{b}_m} \left\{ \sum_{i=1}^m \max_{j(i)} \{I(x_{b_i}; c|x_{b_{j(i)}})\} \right\}, \quad (5.14)$$

where $\mathbf{b}_m = \{b_1, b_2, \dots, b_m\}$ is an subset of m integers out of $1, 2, \dots, n$.

Equivalently, our objective function is:

$$\mathbf{b}_m^* = \operatorname{argmax}_{\mathbf{b}_m} \left\{ \sum_{i=1}^m I(x_{b_i}; c|x_{b_{j(i)}}), 0 \leq j(i) < i. \right\} \quad (5.15)$$

We call this as Maximal-Conditional-Mutual-Information (MCMI) scheme. As we only need to estimate the joint probability mass function of three variables only, in which $x_{b_i}, x_{b_{j(i)}}$ are binary, the computational cost is low. It is worth noting that our method is different from $\max_i \{\min_j I(x_i; c|x_j)\}$ proposed in [4]. They implicitly assume that $I(x_i; c; \mathbf{x}_k|x_j) < 0$, which does not hold in our case.

It is challenging to seek for a globally optimal solution to MCMI scheme. For simplicity, we use sequential forward search [144]. The proposed algorithm starts with a set of possible point candidates. We initialize the selected point subset as a null set. Denote $\mathbf{s}^i = \{x_j^i\}$ as the

current point subset, and $\mathbf{v}^i = \mathbf{x} - \mathbf{s}^i$ as the available feature pool. Then, we aim to find a point in such a way that:

$$x_{j^*}^{i+1} = \underset{x_j^{i+1} \in \mathbf{v}^i, x_s^i \in \mathbf{s}^i}{\operatorname{argmax}} I(x_j^{i+1}; c | x_s^i). \quad (5.16)$$

Here, we assume the conditional mutual information $I(x_j^1; c | x_s^0) = I(x_j^1; c)$. The proposed incremental MCMI scheme is summarized in Algorithm 2.

Algorithm 2: Proposed incremental MCMI scheme to learn LBP structure

Input: a set of potential candidates $\mathbf{x}$, the desired number of neighbors m

Output: an optimal subset $\mathbf{x}_m$

Initialization: Set the selected feature subset as $\mathbf{s}^0 = \emptyset$ and the feature pool $\mathbf{v}^0 = \mathbf{x}$.

- 1: **for** $i \leftarrow 1 : m$ **do**
 - 2: **for** $j \leftarrow 1 : n - i + 1$ **do**
 - 3: Calculate the conditional mutual information $I(x_j^i; c | x_s^{i-1})$ by Eqn. (5.8), where $x_j^i \in \mathbf{v}^{i-1}, x_s^{i-1} \in \mathbf{s}^{i-1}$.
 - 4: **end for**
 - 5: Choose x_j that maximizes $I(x_j^i; c | x_s^{i-1})$.
 - 6: Update $\mathbf{s}^i \leftarrow \mathbf{s}^{i-1} \cup x_j$
 - 7: Update $\mathbf{v}^i \leftarrow \mathbf{v}^{i-1} - x_j$
 - 8: **end for**
 - 9: $\mathbf{x}_m = \mathbf{s}^m$
-

Image patches at different scales or locations may exhibit totally different characteristics. Thus, the “spatial pyramid” [3, 151] was proposed to preserve the patch-wise location information and capture image characteristics at different scales. Similarly, we propose to derive the LBP structure on a patch-wise basis instead of using a unified one for all patches. Each patch is processed separately. The feature vectors of all patches are concatenated to form the final feature vector.

5.5 Experimental Results

The proposed approach can be used in many applications. In this chapter, we show three examples: learning a single LBP structure for texture classification, a set of patch-wise LBP structures for scene recognition and a spatial-temporal LBP structure for dynamic texture recognition. The proposed approach is compared with approaches that directly extract features from the LBP histogram [2, 3], those utilizing handcrafted LBP structures [19], as well as the state of the art reported in solving these problems.

5.5.1 Comparison in Approximating $I(\mathbf{x}_m; c)$

We first show that we derive a closer approximation to high-order mutual information $I(\mathbf{x}_m; c)$. Max-Min Conditional Mutual Information scheme (MmCMI) [4] is similar to the proposed one. Thus, we compare it with the proposed scheme in approximating $I(\mathbf{x}_m; c)$.

We use the 21-land-use dataset [149] for illustration. It contains 21 classes of aerial orthoimagery, and each class has 100 images of resolution 256×256 pixels. We randomly choose 80 images from each class as the training set, and randomly select 8 points from 24 point candidates as the LBP structure. We calculate the ground-truth value of high-order mutual information $I(\mathbf{x}_m; c)$ using this 8-bit LBP structure. Then, we approximate $I(\mathbf{x}_m; c)$ using $\max\{\sum_{i=1}^m I(x_{b_i}; c | x_{b_{j(i)}})\}, 0 \leq j(i) < i$ for the proposed MCMI scheme, and using $\sum_{i=1}^m \max_i\{\min_{j(i)}\{I(x_{b_i}; c | x_{b_{j(i)}})\}\}, 0 \leq j(i) < i$ for MmCMI [4]. We repeat it 100 times, and plot the ground-truth and approximated values of $I(\mathbf{x}_m; c)$ in Fig. 5.3.

We can observe from Fig. 5.3 that even we greedily maximize $I^t(\mathbf{x}_m; c)$ through the proposed incremental MCMI, $I^t(\mathbf{x}_m; c) < I(\mathbf{x}_m; c)$. Thus, $I^t(\mathbf{x}_m; c)$ is a lower bound rather than an upper bound of $I(\mathbf{x}_m; c)$. The proposed MCMI achieves a tighter bound than MmCMI [4]. It justifies our Max-Max scheme as defined in Eqn. (5.14), rather than Max-Min scheme [4].

To compensate possible scaling factor for $I^t(\mathbf{x}_m; c)$, we calculate the correlation coefficients between the approximated value $I^t(\mathbf{x}_m; c)$ and the ground-truth value $I(\mathbf{x}_m; c)$ in 100 trials, and show them in Table 5.1. The proposed MCMI scheme exhibits larger correlation coefficient than MmCMI. It suggests that the proposed MCMI scheme varies more closely with high-order mutual information $I(\mathbf{x}_m; c)$ than MmCMI. In the rest of the experiments, we will utilize the proposed approach to derive the LBP structures using both MCMI and MmCMI schemes, and show their performances for different applications.

Table 5.1: Correlation coefficient between $I^t(\mathbf{x}_m; c)$ and $I(\mathbf{x}_m; c)$ for MmCMI and the proposed MCMI scheme.

Method	Correlation Coefficient
MmCMI [4]	0.20
Proposed MCMI	0.47

Figure 5.3: The proposed MCMI achieves a closer approximation to high-order mutual information $I(\mathbf{x}_m; c)$ than MmCMI [4]. The proposed MCMI is a tighter lower bound of $I(\mathbf{x}_m; c)$.

5.5.2 Texture Classification on the KTH-TIPS-2a Dataset

We now compare the proposed approach with others on the KTH-TIPS-2a dataset [152] for texture classification using basic experimental settings. This dataset contains 11 classes, and each class has four samples (groups). In each sample group, there are 72-108 images taken at different scales from different orientations. We follow the same experimental settings as in [2]. Three samples of each class are used for training and the fourth for testing. We report average classification rate over four random partitions. Same as in [2], we use simple 3-nearest-neighbor classifiers with χ^2 distance metric. Better results using more sophisticated classifiers such as SVM are available in [152].

We compare the proposed approach with LQP [2], which is an alternative to reduce the LBP feature dimensionality by directly extracting features from a large pool of histogram bins. Many handcrafted structures were proposed in LQP [2], among which $Disc_5^{3*}$ performs best. $Disc_5^{3*}$ means features extracted in a disc-shape region of 5×5 pixels and $3*$ means split ternary coding [1]. For a fair comparison to LTP and $Disc_5^{3*}$ LQP, a variant of the proposed approach is used, i.e. after deriving the LBP structure, we use it to generate split ternary codes, similarly as in [1, 2]. We use the default setting as in [1] for LTP, e.g. the threshold $t = 5$. We also utilize the proposed approach to extract LTP features using MmCMI scheme [4]. The results are summarized in Table 5.2. The results for LBP, LTP, $Disc_5^{3*}$ LQP, CWLD [153] and WLD [153] are taken from [2].

We can see that the proposed approach outperforms LTP by more than 10%, which shows that the proposed approach is able to derive a structure that has more potential to capture the intrinsic image structures than the handcrafted one. Compared with $Disc_5^{3*}$ LQP that directly extracts features from the LBP-histogram bins, the proposed approach with MCMCMI scheme improves the classification rate from 64.2% to 71.1%.

Table 5.2: Comparisons with the other approaches on the KTH-TIPS-2a dataset for texture classification.

Method	Recognition Rate
CWLD	56.5%
WLD	59.4%
LBP	58.7%
LTP	60.7%
$Disc_5^{3*}$ LQP [2]	64.2%
Proposed approach with MmCMI	70.2%
Proposed approach with MCMI	71.1%

5.5.3 Scene Recognition on the 21-Land-Use Dataset

We conduct experiments on the 21-land-use dataset [149] for scene recognition using the same experimental setup as in [149, 154]. The spatial pyramid [3, 151] is utilized. Each image is divided into 1 patch for level 0, 5 patches for level 1 and 25 patches for level 2, respectively. For each class, we randomly split it into five equal-size sets. Four of the sets are used for training and the held-out set is used for testing. We use CENTRIST feature [3] as the baseline algorithm. As CENTRIST utilizes 8-bit LBP, we derive LBP structures of 8 points for a fair comparison. The learned LBP structures for the first patch of each pyramid level of trial 1 are shown in Fig. 5.4. These LBP structures are significantly different from each other. For each patch, the intrinsic image characteristic is different, and hence a different LBP structure is needed.

Then, LBP histograms are generated using these 8-bit patch-wise LBP structures. The concatenated feature vector of 31 patches is of size $256 \times 31 = 7936$. To derive a compact feature representation, PCA is applied on each patch to reduce the dimensionality from 256 to 50. Image statistics are shown helpful for scene recognition [3], and hence included as features. For CENTRIST, different weights are assigned to features of different pyramid levels. In our experiments, we use $w_0 = 2.4$, $w_1 = 1.2$, $w_2 = 1$ for level 0, 1, 2, respectively.

The proposed approach is also compared with $Disc_5^{3*}$ LQP [2]. We implement $Disc_5^{3*}$ LQP

Figure 5.4: The learned LBP structures for the first trial of 21-land-use dataset.

according to [2], in which unsupervised k-means algorithm is used to cluster LBP-histogram bins into visual words and Linear SVM is used for classification. For a fair comparison, we also include the results for $Disc_5^{3*}$ LQP using RBF SVM. The experimental results are summarized in Table 5.3.

Table 5.3: Comparisons with the state of the art on the 21-land-use dataset for scene recognition.

Method	Recognition Rate
SPCK [149]	73.1%
SPCK+ [149]	76.1%
SPCK++ [149]	77.3%
BRSP [154]	77.8%
CENTRIST [3]	85.9%
$Disc_5^{3*}$ LQP Linear SVM [2]	83.0%
$Disc_5^{3*}$ LQP RBF SVM [2]	85.6%
Proposed approach with MmCMI	87.4%
Proposed approach with MCMI	88.2%

The proposed approach extracts features in the same 5×5 neighborhood, whereas it outperforms $Disc_5^{3*}$ LQP by 2.6%. It demonstrates the advantages of the proposed approach over directly extracting features from LBP histogram. CENTRIST feature achieves a recognition rate of 85.9%. The proposed approach outperforms it by 2.3%, which shows that the learned

LBP structures can better capture the image characteristics than handcrafted structures. Clearly, the proposed approach to extract LBP features demonstrates a superior performance over others. The confusion matrix of the proposed approach with MCMI scheme is given in Fig. 5.5.

5.5.4 Scene Recognition on the 8-Event Dataset

The 8-event dataset [155] is composed of eight sport classes. Each class has 137 to 250 high-resolution images (from 800×600 to thousands of pixels per dimension). To capture the image micro-structures at the same scale, we resize the images so that the minimum image dimension (height or width) is 600 pixels. We use the same setup as in [3, 154, 155]. For each class, we randomly select 70 images for training and 60 for testing. The experiments are repeated 5 times. PCA is applied to reduce the dimensionality of LBP histogram from 256 to 40, same as in [3].

Figure 5.5: Confusion matrix of the proposed approach with MCMI scheme for the 21-land-use dataset. Only entries with count at least 3 are shown.

The other parameters are the same as for the 21-land-use dataset.

The learned LBP structures are shown in Fig. 5.6. These structures are significantly different from those in Fig. 5.4. As the discriminative information for different applications may not be the same, different LBP structures are needed. The comparisons with the state of the art are summarized in Table 5.4. The baseline algorithm CENTRIST built upon handcrafted LBP structure achieves a recognition rate of 78.3%. The proposed approach using MCMI scheme significantly improves it to 84.3%. It also outperforms $Disc_5^{3*}$ LQP, which achieves a recognition rate of 78.9% using linear SVM and 77.8% using RBF SVM. The confusion matrix of the proposed approach with MCMI scheme is shown in Fig. 5.7.

Table 5.4: Comparison with the state of the art on the 8-event dataset for scene recognition.

Method	Recognition Rate
Scene/Object Model + SIFT [155]	73.4%
RSP + Optimal Selection [154]	77.9%
RSP + Boosting [154]	79.6%
CENTRIST [3]	78.3%
$Disc_5^{3*}$ LQP linear SVM [2]	78.9%
$Disc_5^{3*}$ LQP RBF SVM [2]	77.8%
Proposed approach with MmCMI	83.0%
Proposed approach with MCMI	84.3%

Figure 5.6: The learned LBP structures for the first trial of 8-event dataset.

Figure 5.7: Confusion matrix of the proposed approach with MCMI scheme for the 8-event dataset. Only entries with count at least 3 are shown.

5.5.5 Dynamic Texture Recognition on the DynTex++ Dataset

Dynamic texture is sequences of images of moving scenes that exhibit certain stationarity properties in time [5, 150]. The dynamics of texture elements are statistically similar and temporally stationary. The recognition of DT involves the analysis of both the spatial appearance of static texture patterns and temporal variations in appearance.

The DynTex++ dataset proposed in [23] aims to provide a rich benchmark for DT recognition. It consists of 36 classes of DTs. Each class contains 100 sequences of $50 \times 50 \times 50$ pixels. It mainly contains tightly cropped sequences. Thus, we treat each DT sequence as a whole and do not divide it into patches. We use the same setup as in [5, 23]. For each class, 50 sequences are randomly chosen as the training set, and the other 50 sequences are used in testing. The experiments are repeated 5 times and the average performance is reported. We use 26 binarized pixel differences as potential candidates, as shown in Fig. 5.2(b). The learned spatial-temporal LBP structures for the DynTex++ dataset are shown in Fig. 5.8. The number indicates the order of the neighbor being selected. We select up to 16 neighbors. The STLBP structures for 5 trials are fairly consistent. Specifically, the same model is built for Trial 1, 2, 3 and 5, and the first 13 neighbors are the same for 5 trials.

	t-1	t	t+1																											
Trial 1, 2, 3, 5	<table><tr><td></td><td>6</td><td>14</td></tr><tr><td></td><td>7</td><td></td></tr><tr><td>9</td><td>8</td><td></td></tr></table>		6	14		7		9	8		<table><tr><td></td><td>5</td><td>13</td></tr><tr><td></td><td></td><td></td></tr><tr><td>10</td><td>3</td><td>16</td></tr></table>		5	13				10	3	16	<table><tr><td></td><td>4</td><td>12</td></tr><tr><td></td><td>1</td><td></td></tr><tr><td>11</td><td>2</td><td>15</td></tr></table>		4	12		1		11	2	15
	6	14																												
	7																													
9	8																													
	5	13																												
10	3	16																												
	4	12																												
	1																													
11	2	15																												
Trial 4	<table><tr><td></td><td>6</td><td></td></tr><tr><td></td><td>7</td><td></td></tr><tr><td>9</td><td>8</td><td>16</td></tr></table>		6			7		9	8	16	<table><tr><td></td><td>5</td><td>13</td></tr><tr><td></td><td></td><td></td></tr><tr><td>10</td><td>3</td><td>15</td></tr></table>		5	13				10	3	15	<table><tr><td></td><td>4</td><td>12</td></tr><tr><td></td><td>1</td><td></td></tr><tr><td>11</td><td>2</td><td>14</td></tr></table>		4	12		1		11	2	14
	6																													
	7																													
9	8	16																												
	5	13																												
10	3	15																												
	4	12																												
	1																													
11	2	14																												

Figure 5.8: The learned spatial-temporal LBP structures of the DynTex++ dataset for 5 trials.

Figure 5.9: The average recognition rate over 5 trials for the proposed approach built using different number of neighbors.

There is one free parameter, *i.e.* the desired number of neighbors m . The proposed approach is scalable. We could vary m to obtain different LBP structures according to the requirements on computational complexity and accuracy. The respective LBP structures for different m can be easily derived from Fig. 5.8. In contrast, the handcrafted structure is not scalable. It is difficult to define a consistent rule to build the handcrafted structure for different m . The average recognition rates over 5 trials vs. m are shown in Fig. 5.9. The highest recognition rate is 95.9% when $m = 16$. The recognition rate does not increase significantly after $m = 10$.

It is not feasible to enumerate $2^{26} = 67,108,864$ bins if all 26 neighbors are used to build the LBP histogram. This is exactly the case that direct feature selection/extraction from histogram

bins is not applicable. Alternatively, we compare the proposed approach with LBP-TOP [19], in which the large LBP structure is broken into small ones. In LBP-TOP, LBP features are extracted from three orthogonal planes, *i.e.* XY-plane (spatial LBP), XT-plane and YT-plane [19]. The feature vectors of three planes are concatenated to form the final feature vector. We summarize the performance comparisons with the state of the art in Table 5.5.

Table 5.5: Comparison with the state of the art on the DynTex++ DT dataset.

Method	Recognition Rate
DL-PEGASOS [23]	63.7%
Dynamic fractal analysis (DFS) [5]	89.9%
LBP-TOP [19]	93.2%
Proposed approach with MmCMI	95.7%
Proposed approach with MCMI	95.9%

We implement and test LBP-TOP on the Dyntex++ dataset, which achieves a recognition rate of 93.2%. The proposed approach with MCMI scheme improves the recognition rate to 95.9%, which demonstrates a superior performance to other approaches.

5.5.6 Dynamic Texture Recognition on the UCLA Database

The UCLA dynamic texture database [156, 157] has been widely used as a benchmark dataset for DT categorization [5, 23, 158, 159]. It consists of 50 classes of DTs, each with 4 sequences captured from different viewpoints. There are several breakdowns when evaluating the dataset [5]. Among them, we choose the following three representative settings:

50-Class: This is the original setting for the UCLA dataset. 4-fold cross-validation is used. The average recognition rate over 4 trials is reported. It is rather a simple setting and a recognition rate of 100% is achieved in [5].

9-Class: Those 50 classes can be clustered to 9 classes by combining the sequences of different viewpoints. We use the same experimental settings as in [5, 23]. We randomly choose half of

the dataset as the training set and the other half as the testing set. The experiments are repeated 10 times and the average recognition rate is reported. This setting is challenging and used to evaluate DT recognition under viewpoint changes.

Shift-invariant Recognition(SIR)-class: In [159], each original sequence is spatially partitioned into non-overlapping left and right halves and 400 sequences are obtained. The “shift-invariant recognition” [159] was implemented to compare the sequences only between halves to test the shift-invariant property. This setting is very challenging. Thus, in general rank-5 recognition rate was reported [5, 159].²

We try different ways to divide DTs into patches. The best performance is achieved using the following settings: for 50-class and SIR, we divide DTs along time axis into two patches of equal size; for 9-class, we spatially divide the sequences into 3×3 patches. We use the same potential candidates as shown in Fig. 5.2(b). As the number of training samples is limited, to avoid over-fitting we train one 12-neighbor STLBP structure for all patches. Nearest-neighbor classifier with χ^2 distance is utilized. The proposed approach is compared with published methods in literature [5, 23, 159], as well as LBP-TOP [19] built upon handcrafted structures. The results are summarized in Table 7.5.

Table 5.6: Comparison with the state of the art on the UCLA DT dataset for 50-class, 9-class and SIR settings.

Method	50-Class	9-Class	SIR
Distributed spacetime orientation [159]	81.0%	-	60.0%
DL-PEGASOS [23]	99.0%	95.6%	-
DFS [5]	100.0%	97.5%	73.8%
LBP-TOP [19]	87.5%	85.8%	93.8%
Proposed approach with MmCMI	99.5%	98.4%	92.3%
Proposed approach with MCMI	100.0%	98.4%	94.5%

²If among top-5 matches there is at least one gallery sample with the same label as the probe sample's, it is counted as a successful recognition. The recognition rate calculated in this way is rank-5 recognition rate.

Compared with previous results [5, 23, 159] in literature, the proposed approach demonstrates a superior performance. On 9-class setting, it reduces the error rate from 2.5% to 1.6%. On the most challenging SIR setting, it improves the recognition rate from 73.8% to 94.5%. Compared with LBP-TOP [19], the proposed approach achieves a much better performance on 50-class and 9-class setting, and a slightly better performance on SIR setting. The performance of LBP-TOP on SIR setting is also significantly better than previous results. This is partially because LBP-TOP feature is less affected by shift operation, especially the LBP histograms extracted from XT-plane and YT-plane.

5.5.7 Analysis of Computational Complexity

We analyze the computational complexity in both off-line training stage and online testing stage. In the training stage, compared with handcrafted LBP/LTP, we need an additional step to estimate the conditional mutual information $I(x_i; c|x_j)$ in order to determine the LBP structure. To derive $I(x_i; c|x_j)$, we need to estimate the joint probability mass function (PMF) $p(x_i, x_j, c)$. When incremental search is utilized, we need to estimate roughly $n(m-1)$ such PMFs, in which x_i, x_j are binary. Each PMF corresponds to a histogram of $4N_c$ bins, where N_c is the number of classes. The computational complexity is $O(mnN_c)$. Two points are worth to mention. Firstly, we only need to store the estimated histogram of a class, but not histograms of all samples in this class. Thus, much less memory is required, which is particularly important when dealing with a large-scale database. Secondly, the computational complexity grows linearly with the number of candidates n and the desired number of neighbors m . In general, we expect $m \leq 24$ for binary encoding and $m \leq 16$ for ternary encoding. Thus, ideally we could utilize a large number of potential candidates, and hence deal with high-dimensional LBP features. For those applications we tested, we find that $n = 24$ for spatial LBP and $n = 26$ for spatial-temporal LBP yield a satisfactory performance. In the testing stage, the LBP structures derived in the training stage can be used in the same way as handcrafted ones. Thus, the computational complexity is

the same as handcrafted LBP/LTP.

We summarize the time cost required for different algorithms on the 21-land-use dataset and 8-event dataset in Table 5.7. For simplicity, we show the total time required for both training and testing. We use Matlab 2012b on Intel Core 2 Duo CPU with 4Gb memory. CENTRIST [3] utilizes handcrafted structures, which can be viewed as the baseline time consumption. The proposed approach needs additional time to derive the LBP structures. Even so, the time required is still reasonable. We also include the time cost of LQP for reference. We use FKMEANS package [160] for k-means clustering used in LQP.

Table 5.7: Comparisons of time consumption for CENTRIST, $Disc_5^{3*}$ LQP and the proposed approach on the 21-land-use dataset and the 8-event dataset.

Method	21-land-use (s)	8-event (s)
CENTRIST [3]	135.1	439.7
$Disc_5^{3*}$ LQP [2]	9694.6	5293.9
Proposed approach	1580.6	2992.6

5.6 Summary

In general, LBP features using more bits improve the performance, at the cost of high feature dimensionality. The dimensionality easily goes beyond the capability of approaches that directly extract features from the histogram bins as it increases exponentially with the number of LBP bits. Alternatively, the large structure was broken into small handcrafted ones. However, handcrafted LBP may not be optimal. Instead, we learn discriminative LBP structures from image and video data. The LBP structure learning is casted as a point selection problem, where the optimal subset of neighboring pixels rather than histogram bins are selected. Due to the high dependency among image pixels in a local neighborhood, existing feature-selection algorithms may not be suitable in our scenario. We thus propose a novel feature-selection method

by approximating the high-order mutual information with a set of low-order conditional mutual information, and achieving Max-Dependency via maximizing the approximated one. As shown in the experiments, our proposed incremental MCMI scheme can well solve the LBP-structure-learning problem. Moreover, the proposed approach is readily incorporated to spatial pyramid framework that can better capture the intrinsic characteristics of image patches at different locations and scales. The proposed approach outperforms the state-of-the-art solutions to scene and dynamic texture recognition significantly on several benchmarked datasets.

Chapter 6

Principal Histogram Analysis of LBP and its Applications in Visual Recognition

LBP features are noisy as many bins are not reliably estimated. Furthermore, useful bins and noisy bins may be different for different tasks. In this chapter ¹, Principal Histogram Analysis is utilized to remove the unreliable information that resides in LBP features. Consequently, two learning-based LBP features are proposed. To improve the reliability of LBP histogram, a “Super Histogram” is proposed for dynamic texture recognition. The proposed approaches achieve a superior performance on various recognition tasks.

6.1 Introduction

LBP has been widely used in various applications, but its performance is limited by the reliability of LBP histogram. The problem mainly arises due to the limited number of elements in the histogram. Firstly, in order to preserve the locality, the image is divided into patches and a histogram is extracted for each patch. This leads to fewer elements in each histogram. For example in face recognition, the image of size 130×150 is divided into $7 \times 7 = 49$ patches and each patch has around 400 pixels. Averaging on 256 bins, each bin has less than 2 ele-

¹This work is partially published in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2013 (conference paper [4] in author’s publication list).

ments. Statistically, the occurrence probability of LBP cannot be reliably estimated. Secondly, the image is smooth in nature, and hence some patterns rarely appear in the histogram. This will lead to uneven distribution of LBP features. The occurrence probability of those rare patterns cannot be well estimated. These patterns are less reliable and harmful to reliable classification. In addition, LBP histogram may be corrupted by image noise.

In this chapter, two learning-based approaches are proposed to tackle the reliability issues of LBP features. As pointed out in [24, 25], PCA can be used not only for dimension reduction, but more importantly to remove the dimensions that are harmful to reliable classification. In the proposed approaches, Principal Histogram Analysis (PHA) is applied on the covariance matrix of LBP histogram to remove the unreliable information residing in LBP histogram. Only those reliable dimensions corresponding to large eigenvalues are preserved. It is different from the previous approaches [21, 22], in which PCA is applied on the concatenated LBP feature vector. Those approaches [21, 22] suffer from the small-sample-size problem. If the image is divided into 49 patches, it will generate a concatenated feature vector of $49 \times 256 = 12544$ dimensions for $LBP_{8,2}$. It is very difficult to reliably estimate the covariance matrix of such a high dimensionality. In addition, 12544 dimensions are too large to handle. In contrast, in the proposed approaches, the feature dimensionality is 256 only.

For dynamic texture recognition, LBP features are used to capture spatial information in previous approach [23]. One LBP histogram is extracted for each frame, and the histograms are compared on a frame-to-frame basis. Dynamic texture recognition largely relies on spatial information. It is consistent with the human perception on dynamic texture. Fig. 6.1 shows some samples of dynamic textures.

Given only one frame per dynamic texture, human can easily differentiate different dynamic textures. The temporal information serves as further verification. In addition, the frames of dynamic texture are in general similar to each other. Inspired by these, a “Super Histogram” is constructed by averaging (or summing up) over all the frames along temporal direction. Tem-

Figure 6.1: Samples of dynamic textures from DynTex database.

poral information is partially transferred to the “Super Histogram”. The proposed “Super Histogram” consists of more elements, and hence can be better estimated. The “Super Histogram” of the proposed learning-based LBP features demonstrates a superior performance on dynamic texture recognition compared with the state-of-the-art methods.

6.2 Related Work

Many approaches have been proposed to remove the unreliable dimensions of LBP features. Uniform-LBP [16] suppresses the unreliable information by grouping non-uniform patterns into one histogram bin. However, discriminative information is lost during grouping. More importantly, useful patterns and noisy patterns are different for each application. The problem of uneven distribution of LBP was addressed in [83]. However, when PCA is applied on the concatenated feature vector, the high dimensionality leads to classical small-sample-size (SSS) problem. In [21, 22], similarly principal component analysis was applied on the concatenated

feature vector. To avoid SSS problem, in CENTRIST feature [3] principal component analysis was applied on the LBP histogram of 256 bins only built from a 3×3 neighborhood. As LBP is sensitive to quantization noise, a tri-state local ternary pattern (LTP) was proposed in [161]. However, the number of bins increases dramatically, e.g. $LTP_{8,2}$ has $3^8 = 6561$ bins. Although the number of bins can be reduced by splitting LTP code into positive and negative LBP codes, it is still doubled from LBP.

6.3 The Proposed Learning-Based LBP

In many applications, an image is often divided into patches and one LBP histogram is extracted from each patch in order to preserve locality. Then, these histograms are concatenated to form the final feature vector. These histograms are often not reliably estimated due to insufficient elements in LBP histogram. Thus, two learning-based approaches are proposed to remove the unreliable dimensions in LBP features. Many dimension-reduction techniques can be utilized [6, 7, 12, 25]. In the proposed approaches, Principal Histogram Analysis is applied on the LBP histogram of each patch. This dimension is much smaller than the dimension of the concatenated feature vector in [21, 22].

The proposed learning-based LBP can be constructed in a patch-independent manner or a patch-dependent manner. For patch-independent learning-based LBP (PI-LBP), all covariance matrices of different patches are summed up so that the resulting covariance matrix can be better estimated. However, patch locality is discarded as the histograms of different patches are treated as the same. In order to preserve locality, a patch-dependent learning-based LBP (PD-LBP) can be constructed, i.e. one projection matrix is built for each patch.

6.3.1 The Proposed Patch-Independent Learning-Based LBP

In some applications such as texture recognition or dynamic texture recognition, the histogram of each patch is similar. Thus, the patch locality is not critical, and hence the covariance matrices of different patches can be summed up so that the unreliable information in the histogram can be better removed. In this section, the proposed Patch-Independent learning-based LBP is derived by applying principal component analysis on the LBP histogram. The block diagram to build PI-LBP is shown in Fig.6.2. Let $\mathbf{h}_{i,j,k} \in \mathbb{R}^{256}$ denote the histogram of i -th patch of j -th image of k -th class. The total covariance matrix is obtained as:

$$\Sigma_t = \sum_i \sum_{j,k} (\mathbf{h}_{i,j,k} - \mu_i)(\mathbf{h}_{i,j,k} - \mu_i)^T, \quad (6.1)$$

where $\mu_i = \frac{1}{N} \sum_{j,k} \mathbf{h}_{i,j,k}$ is the mean histogram of i -th patch and N is the total number of training samples. PCA is then applied on Σ_t , e.g. $\Sigma_t = \Phi \Lambda \Phi^T$. Eigenvectors Φ_m corresponding to m largest eigenvalues are selected, and $\mathbf{h}_{i,j,k}$ is projected to the feature space as $\mathbf{f}_{i,j,k} = \Phi_m^T \mathbf{h}_{i,j,k}$.

Mahanalobis distance is shown to perform well on classification. Thus, we normalize the feature vector by the inverse of the within-class scatter matrix Σ_w^f , which is calculated as:

$$\Sigma_w^f = \sum_i \sum_k \sum_j (\mathbf{f}_{i,j,k} - \mu_{i,k}^f)(\mathbf{f}_{i,j,k} - \mu_{i,k}^f)^T, \quad (6.2)$$

Figure 6.2: Block diagram of patch-independent learning-based LBP. Covariance matrices of all patches are summed up and one unified projection matrix P is obtained for all patches.

where $\mu_{i,k}^f = \frac{1}{N_k} \sum_j \mathbf{f}_{i,j,k}$ is the mean feature vector of i -th patch of k -th class in the feature space and N_k is the number of samples for k -th class. After performing PCA on Σ_w^f , e.g. $\Sigma_w^f = \Phi_w \Lambda_w \Phi_w^T$, the normalized feature vector for PI-LBP is obtained as:

$$\mathbf{g}_{i,j,k} = (\Phi_m \Phi_w \Lambda_w^{-0.5})^T \mathbf{h}_{i,j,k}. \quad (6.3)$$

Discriminant analysis can be applied to further reduce the dimensionality. The between-class scatter matrix is calculated based on the normalized feature vector:

$$\Sigma_b^g = \sum_i \sum_k (\mu_{i,k}^g - \mu_i^g)(\mu_{i,k}^g - \mu_i^g)^T, \quad (6.4)$$

where $\mu_{i,k}^g = \frac{1}{N_k} \sum_j \mathbf{g}_{i,j,k}$, $\mu_i^g = \frac{1}{c} \sum_k \mu_{i,k}^g$ and c is the number of class. Then, we apply PCA on Σ_b^g , e.g. $\Sigma_b^g = \Phi_b \Lambda_b \Phi_b^T$. The first t eigenvectors $\Phi_{b,t}$ corresponding to t leading eigenvalues are selected and the final feature vector is obtained as:

$$\hat{\mathbf{g}}_{i,j,k} = (\Phi_m \Phi_w \Lambda_w^{-0.5} \Phi_{b,t})^T \mathbf{h}_{i,j,k}. \quad (6.5)$$

In such a way, the feature dimensionality is further reduced.

For face recognition and dynamic texture recognition, the nearest-neighbor classifier with Chi-squared distance is used. Since $\mathbf{g}_{i,j,k}$ or $\hat{\mathbf{g}}_{i,j,k}$ may be negative, Chi-squared distance is modified slightly as:

$$\chi_w^2(\mathbf{x}, \mathbf{y}) = \sum_{i,j} w_j \frac{(x_{i,j} - y_{i,j})^2}{|x_{i,j}| + |y_{i,j}|}, \quad (6.6)$$

where $\mathbf{x}, \mathbf{y}$ are concatenated LBP feature vectors of two samples, $x_{i,j}$, $y_{i,j}$ are i -th dimensions of j -th patch and w_j is the weight for patch j . For scene recognition, a support-vector machine with RBF kernel is used for classification.

6.3.2 The Proposed Patch-dependent Learning-Based LBP

In some applications such as face recognition, the histogram of each patch is significantly different from each other. PD-LBP will be able to extract intrinsic image characteristics of each patch. The block diagram of PD-LBP is shown in Fig. 6.3.

For PD-LBP, the total scatter matrix for i -th patch is defined as:

$$\tilde{\Sigma}_i = \sum_{j,k} (\mathbf{h}_{i,j,k} - \mu_i)(\mathbf{h}_{i,j,k} - \mu_i)^T. \quad (6.7)$$

PCA is then applied on $\tilde{\Sigma}_i$, e.g. $\tilde{\Sigma}_i = \tilde{\Phi}_i \tilde{\Lambda}_i \tilde{\Phi}_i^T$. We select $\tilde{m}$ eigenvectors $\tilde{\Phi}_{i,\tilde{m}}$ corresponding to $\tilde{m}$ largest eigenvalues and project the histogram of i -th patch $\mathbf{h}_{i,j,k}$ to the feature space as $\tilde{\mathbf{f}}_{i,j,k} = \tilde{\Phi}_{i,\tilde{m}}^T \mathbf{h}_{i,j,k}$. The rest procedures are similar to that of PI-LBP. In the end, one projection matrix is built for each patch.

6.3.3 Relations to Other Approaches

The proposed PI-LBP is directly comparable to uniform-LBP [16]. ULBP reduces the number of bins from 256 to 59, whereas PI-LBP reduces the number of dimensions per patch from 256 to m . In ULBP, non-uniform patterns are considered as noisy patterns, and hence grouped into one bin. However, discriminative information may be lost during grouping. In addition,

Figure 6.3: Block diagram of patch-dependent learning-based LBP. In order to capture the intrinsic image characteristic of different patches, one projection matrix is built for each patch.

some non-uniform patterns may have higher occurrence frequency than uniform patterns. More importantly, useful and noisy patterns are different for different applications. The proposed approaches can better remove the unreliable information in LBP histogram than ULBP.

Compared with PCA on the concatenated LBP features (PCA-cLBP) [21,22], the covariance matrix of LBP histogram is better estimated in the proposed approaches. If the image is divided into P patches, we have in total NP patches to estimate the covariance matrix of size 256×256 for PI-LBP. For PCA-cLBP, we have only N samples to estimate the covariance matrix of $256P \times 256P$. Apparently much more samples are available to estimate a much smaller covariance matrix for the proposed PI-LBP. Thus, the covariance matrix of PI-LBP can be better estimated. For PD-LBP, we have N samples to estimate the covariance matrix of 256×256 for each patch, which can still be better estimated than that of PCA-cLBP. As one projection matrix is built for each patch, patch locality is preserved and the projection matrix can better capture the intrinsic image characteristic of each patch.

6.4 The Proposed Super Histogram

DT recognition largely relies on the spatial information [23]. For example, on UCLA 50-class classification problem, a recognition rate of 90% is achieved by utilizing spatial LBP alone. It is consistent with the human perception on DTs. Given only one frame per DT, human can easily differentiate DTs. The temporal information serves as further verification. In general, each frame of DT is spatially similar. Inspired by these, a super histogram is proposed.

Let $\mathbf{h}_{i,p} \in \mathbb{R}^{256}$ denote the histogram of i -th patch of p -th frame. The super histogram $\mathbf{h}_i^s$ for i -th patch of this sequence is:

$$\mathbf{h}_i^s = \frac{1}{P} \sum_p \mathbf{h}_{i,p}, \quad (6.8)$$

where P is the number of frames in this sequence. The proposed super histogram of LBP mainly captures the local spatial appearance of DTs. The temporal information is partially transferred into the super histogram, as the super histogram is obtained by averaging the histograms over all frames.

The proposed super histogram can better handle the spatial-temporal variations in DTs than [23], in which the histograms are compared on a frame-to-frame basis. The histogram estimated using each frame is less reliable as it has fewer elements than the super histogram. In addition, due to the temporal variations, it is difficult to match two DTs on a frame-to-frame basis. Alpha-trimmed histogram [162] could yield a better performance when the histograms are noisy, but at a higher computational cost. For simplicity, we only consider super histogram here.

The proposed super histogram is also superior to the approach that a LBP histogram is extracted on the super image. Following the similar concept of super histogram, a super image is obtained by averaging over all the frames along time-axis. For the super image, we end up with only one frame. The histogram built upon the super image has a limited number of elements only, and hence it is still not reliably estimated. In the proposed approach, the image local structures are extracted from each frame, and the structure patterns are used to construct the super histogram. In contrast, these image local structures are lost when constructing the LBP histogram of super image. To illustrate the difference between these two approaches, we plot them for the first patch of the first DT in the UCLA database as shown in Fig. 6.4. Each frame is divided into $3 \times 3 = 9$ patches. For the super histogram, the occurrence probabilities of all the LBP patterns are well estimated. However, for the histogram of super image, the

Figure 6.4: The super histogram and the histogram of super image.

occurrence probabilities of many patterns are zero as shown in Fig. 6.4(b).

6.5 Experiments on Face Recognition

The proposed learning-based PI-LBP and PD-LBP are compared with LBP [17], ULBP [16] and PCA-cLBP [21] on the FERET database [163] and the AR database [132]. We follow the same experimental setup as in [17]. The cropped face image is of size 130×150 , and divided into $7 \times 7 = 49$ patches. One histogram is obtained for each patch. The weighted Chi-squared distance is chosen as the distance measure. The weight used for each patch is the same as in [17].

6.5.1 Experiments on the FERET database

On the FERET database, FA and FB datasets are chosen. It consists of 2390 images of 1195 persons (One image from FA/FB for each person). First 497 persons are used in training (in total 994 images), and the rest are used in testing. Some samples images are shown in Fig. 6.5.

The face recognition rates for PD-LBP, PD-LBP and ULBP are shown in Fig. 6.6. The

recognition rate is 98.28% at 70 dimensions for PI-LBP and 98.71% at 20 dimensions for PD-LBP, which are both slightly better than 97.42% for LBP and 97.13% for ULBP. In terms of speed, the proposed PD-LBP achieves the highest recognition rate at 20 dimensions, which is significantly smaller than 59 dimensions of ULBP and 256 dimensions for LBP. On this dataset, the proposed PD-LBP consistently outperforms PI-LBP. As the face is properly aligned, each image patch represents a significantly different facial part. The PD-LBP can better capture intrinsic property of different patches. PCA-cLBP achieves a recognition rate of 98.85% at 320 dimensions, which is comparable to the proposed approach. However, in terms of speed, the proposed approaches are significantly faster than PCA-cLBP. The complexity of PD-LBP is $O(20 \times 256 \times 49)$, whereas the complexity of PCA-cLBP is $O(320 \times 256 \times 49)$, 16 times of PD-LBP.

6.5.2 Experiments on the AR database

The AR database contains over 4000 color images of 126 persons with different facial expressions, illumination conditions and occlusions. 14 frontal images of 75 subjects are selected.

Figure 6.5: Sample images of the FERET database

Figure 6.6: Plot of the recognition rates for PI-LBP and PD-LBP on the FERET database.

Figure 6.7: Sample images of the AR database

First 7 images of each subject are used in training (in total 525 images), and the rest are used in testing. The images are pre-processed similarly as in [17].

The recognition rates for PI-LBP, PD-LBP and ULBP are shown in Fig. 6.8. The recognition rate is 98.48% at 90 dimensions for PI-LBP and 99.05% at 20 dimensions for PD-LBP, which are both better than 98.10% for LBP and 97.90% for ULBP. By better removing the unreliable information of LBP histogram, the proposed PI-LBP and PD-LBP demonstrate a superior performance to LBP and ULBP in terms of both speed and recognition rate. The proposed PD-LBP also outperforms PCA-cLBP, which achieves a recognition rate of 98.48% at 220 dimensions.

Figure 6.8: Plot of the recognition rate for PI-LBP and PD-LBP on the AR database.

6.5.3 Summary of Face Recognition Results

We summarize the performance comparisons on face recognition in Table 6.1. Compared with LBP and ULBP, the proposed PI-LBP and PD-LBP achieve a superior performance on both databases. PD-LBP achieves a higher recognition rate at a lower dimensionality, as locality is preserved in PD-LBP, and such information is important for face recognition.

Table 6.1: Summary of performance comparisons on face recognition.

Approaches	FERET	AR
LBP	97.42%	98.10%
ULBP	97.13%	97.90%
PCA-cLBP	98.85%	98.48%
PI-LBP	98.28%	98.48%
PD-LBP	98.71%	99.05%

6.6 Experiments on Dynamic Texture Recognition

In this section, we will show that by utilizing the proposed “Super Histogram” and learning-based PI-LBP/PD-LBP, a superior performance is achieved on the UCLA database [156, 157], the DynTex database [164] and the DynTex++ database [23] compared with the state of the art. Our initial experiment shows that $LBP_{8,1}$ yields a better performance, and hence it is chosen as the feature descriptor.

6.6.1 DT Recognition on the UCLA Database

The UCLA DT database [156, 157] has been widely used as a benchmark dataset [5, 23, 158]. It consists of 50 classes of DTs, and 4 sequences for each class. Each sequence contains 75 frames of 48×48 pixels. Some samples are shown in Fig. 6.9. These 50 classes can be further grouped into 9 classes as they contain the same DT at different viewpoints. 9-class classification on the UCLA database is more challenging [23, 158]. Our initial experiments show that on the UCLA database, the recognition rates are insensitive to the choice of the number of patches. Here, each frame is divided into $3 \times 3 = 9$ patches.

For 50-class classification, we gradually increase the number of frames of the testing sequences, e.g. first frame, first two frames, and etc. The LBP histograms of those frames are averaged along temporal direction. For the training sequences, the histograms are averaged over all the frames. We perform 4-fold cross validation as in [5, 23, 158], and the average recognition rates are shown in Table 6.2.

It is interesting to note that by utilizing the first frame only, a high recognition rate is achieved, e.g. 97.5% for PI-LBP and 97.0% for PD-LBP. It is consistent with the human perception that DT recognition largely relies on the spatial information. Furthermore, it enables fast searching or indexing of DTs as one frame per DT only is required. When all the frames

Figure 6.9: Samples of the UCLA database.

Table 6.2: The recognition rates vs. the number of frames in the test sequence.

Num of Frame	LBP	ULBP	PCA-cLBP	PI-LBP	PD-LBP
1	97.5%	95.0%	92.0%	97.5%	97.0%
2	97.5%	94.5%	93.5%	98.0%	98.0%
5	96.5%	95.5%	95.0%	98.0%	97.0%
10	97.0%	96.5%	95.0%	98.0%	98.0%
20	98.0%	97.0%	97.5%	99.5%	97.5%
40	98.0%	98.0%	99.0%	99.0%	99.0%
75	99.5%	99.5%	99.5%	100.0%	100.0%

are utilized, a recognition rate of 100% is achieved for the proposed PI-LBP and PD-LBP.

For 9-class classification, we adopt the same experimental setup as in [5, 23]. The experiment is repeated 20 times. In each trial, each class is randomly bisected. Half are used for training, and the other half are used for testing.

The results are summarized in Table 7.5. A slightly higher recognition rate is achieved for the proposed PI-LBP and PD-LBP compared with LBP, ULBP and PCA-cLBP. The current best recognition rate is 100.0% for 50-class classification and 97.5% for 9-class classification [5]. The super histograms of the proposed PI-LBP and PD-LBP achieve a error-free classification

for 50-class setting and better recognition rates of 98.2% and 98.1% for 9-class classification, respectively.

Table 6.3: The average recognition rates on the UCLA DT database for 50-class and 9-class classification problems.

Method	50-Class	9-Class
Bag of dynamical systems [158]	-	80.0%
DL-PEGASOS algorithm [23]	99.0%	95.6%
Dynamic Fractal Analysis [5]	100.0%	97.5%
LBP + super histogram	99.5%	97.9%
ULBP + super histogram	99.5%	97.8%
PCA-cLBP + super histogram	99.5%	97.7%
PI-LBP + super histogram	100.0%	98.2%
PD-LBP + super histogram	100.0%	98.1%

6.6.2 DT Recognition on the DynTex database

The DynTex database [164] contains more than 650 sequences.² Some samples are shown in Fig. 6.10. For each sequence, the first 250 frames are extracted and down-sampled to 288×352 . The DynTex database is further grouped into three benchmark datasets: Alpha dataset (60 DTs of 3 classes), Beta dataset (162 DTs of 10 classes) and Gamma dataset (264 DTs of 10 classes). The sequences contain dynamic textures in the complex background, and hence more complicated and challenging.

The proposed PI-LBP and PD-LBP are compared with LBP, ULBP and PCA-cLBP on these three datasets. The experiments are repeated 20 times. For each trial, half of each class are used in training and the other half are used in testing. Let N_p denote the number of patches per row/column. The recognition rates vs. N_p on Alpha, Beta and Gamma datasets are shown in Fig. 6.11(a), 6.11(b) and 6.11(c), respectively. When $N_p = 1$, PCA-cLBP is identical to PI-LBP/PD-LBP, and hence the same recognition rate is achieved for these three approaches.

²An old DynTex database that consists of 35 DT sequences was used by researchers [5, 19]. Compared with the old one, the new DynTex database consists of non-cropped sequences. It is far more complete and complicated.

Figure 6.10: Samples of the DynTex database.

Alpha dataset is a relatively simple dataset. By utilizing the super histogram, good recognition rates are achieved for all five approaches. The highest recognition rate is 99.2% achieved by PI-LBP. Beta dataset is more challenging than Alpha dataset. Fig. 6.11(b) shows a significant drop on accuracy compared with Alpha dataset. PD-LBP significantly outperforms other approaches when $N_p = 7$ or $N_p = 10$. The highest recognition rate is 89.1% achieved by PD-LBP. Gamma dataset is the most challenging dataset among three. Fig. 6.11(c) shows that for $N_p \geq 2$, the proposed PD-LBP consistently outperforms other approaches by a significant amount. The highest recognition rate is 87.8% achieved by PD-LBP.

The performance on three datasets of the DynTex database are summarized in Table 6.4. The proposed PI-LBP and PD-LBP consistently outperform LBP, ULBP and PCA-cLBP by a significant amount on all three datasets. On the most challenging Gamma dataset, the proposed PD-LBP improves the recognition rate by 2.6% compared with PCA-cLBP.

Figure 6.11: The average recognition rates vs. the number of patches per row/column on the DynTex database.

Table 6.4: Performance comparisons on Alpha, Beta and Gamma datasets of the DynTex database.

Method	Alpha Dataset	Beta Dataset	Gamma Dataset
LBP + super histogram	97.3%	86.1%	79.6%
ULBP + super histogram	97.0%	85.3%	77.2%
PCA-cLBP + super histogram	98.0%	85.0%	85.2%
PI-LBP + super histogram	99.2%	87.7%	85.4%
PD-LBP + super histogram	98.5%	89.1%	87.8%

6.6.3 DT Recognition on the DynTex++ Database

In order to provide a richer benchmark for DT recognition, the DynTex database [164] is re-organized as the DynTex++ database in [23]. It consists of 36 classes of DTs. Each class

Figure 6.12: Samples of the DynTex++ database.

contains 100 sequences of size $50 \times 50 \times 50$. We use the same experimental setting as in [5, 23]. For each trial, 50 sequences are randomly selected from each class as the training and gallery set, and the other 50 sequences are used in testing. The experiment is repeated 20 times. The average recognition rates vs. N_p are shown in Fig. 6.13.

The recognition rates decrease as N_p increases. The DynTex++ database is built from tightly cropped sequences, and only one DT in each sequence. The image patches are similar to each other, and hence the patch locality is less important. By dividing the image into patches, the histogram has less elements, and hence is less reliable. The highest recognition rate is achieved when $N_p = 1$ for the DynTex++ dataset. When $N_p = 1$, PCA-cLBP is identical to PI-LBP and PD-LBP.

The proposed PI-LBP and PD-LBP are also compared with the state-of-the-art solutions to dynamic texture recognition. The previous best recognition rate is 89.9% achieved by DFS [5]. Before that, the recognition rate is 63.7% for DL-PEGASOS [23]. As shown in Table 6.5, the proposed super histogram of PI-LBP/PD-LBP significantly improves the recognition rate from 89.9% to 91.9%. When a SVM with a RBF kernel is used for classification, the recognition rate further increases to 92.4%.

The recognition rates of each class for the proposed approaches and DFS approach [5] are

Figure 6.13: The average recognition rates of LBP, ULBP, PCA-cLBP, PI-LBP and PD-LBP vs. N_p on the DynTex++ database.

Table 6.5: The average recognition rates on the DynTex++ DT database.

Method	Recognition Rate
DL-PEGASOS [23]	63.7%
Dynamic fractal analysis (DFS) [5]	89.9%
LBP + super histogram + 1-NN	89.4%
ULBP + super histogram + 1-NN	88.5%
PCA-cLBP/PI-LBP/PD-LBP + super histogram + 1-NN	91.9%
PCA-cLBP/PI-LBP/PD-LBP + super histogram + RBF SVM	92.4%

shown in Fig. 6.14. In [5], mainly the temporal variations are exploited, whereas in the proposed approach, mainly the spatial variations are exploited. The recognition rates of each class are significantly different for these two approaches, which indicates that they are capable to handle different DTs. It will be advantageous to fuse these two approaches.

6.7 Experiments on Scene Recognition

We also evaluate the proposed PI-LBP and PD-LBP for scene recognition on the 15-class scene category database. The database was collected gradually by several research groups [151, 165,

Figure 6.14: The recognition rates of the proposed approach and DFS [5] for each class of the DynTex++ database.

166]. It consists of various indoor and outdoor scenes: office, store, coast and etc. Each scene category contains 216 to 410 images of resolution around 300×250 . We follow the same experimental setup as in [167]. For each trial, 100 images are randomly selected from each category and used in training, and the rest are used in testing. The “spatial pyramid” [151, 167] is utilized. Each image is divided into 25 blocks (level 2), 5 blocks (level 1) and 1 block (level 0), respectively. SVM with a RBF kernel is used for classification. For PD-LBP and PI-LBP, the dimension per block is 80. $LBP_{8,1}$ is utilized as the feature descriptor.

We summarize the results in Table 6.6. The average recognition rate of the proposed PD-LBP and PI-LBP over 5 trials is $83.21\% \pm 0.64\%$ and $82.65\% \pm 0.46\%$, respectively. The average recognition rate for LBP and ULBP is $76.32\% \pm 0.50\%$ and $79.32\% \pm 0.31\%$, respectively. The proposed approaches outperform LBP and ULBP. CENTRIST [168] and its variant spatial PACT [167] are recently developed for scene recognition. We download the source code of spatial PACT [167] and repeat the experiment on the same dataset. The average recognition rate for spatial PACT is $82.53\% \pm 0.36\%$. The proposed PI-LBP and PD-LBP achieve a comparable

Figure 6.15: Sample images of the 15-class scene database.

performance to CENTRIST.

Table 6.6: Performance comparisons on the 15-scene database.

Method	Recognition Rate
LBP	76.32% $\pm$ 0.50%
ULBP	79.32% $\pm$ 0.31%
CENTRIST	82.53% $\pm$ 0.36%
PI-LBP	82.65% $\pm$ 0.46%
PD-LBP	83.21% $\pm$ 0.64%

As claimed in [168], CENTRIST and spatial PACT have a clear advantage in recognizing place and scene image, but they perform poor for object recognition. On face recognition, the recognition rate for spatial PACT using nearest-neighbor classifier is 63.61% on FERET database and 92.00% on AR database. If using linear SVM classifier, the recognition rate of spatial PACT is 60.60% on FERET database and 98.10% on AR database. In contrast, the proposed PI-LBP and PD-LBP perform well for a wide range of applications.

6.8 Summary

In this chapter, by utilizing Principal Histogram Analysis, PI-LBP and PD-LBP are proposed to tackle the reliability issues of LBP histogram. By removing the unreliable information in LBP histogram, the proposed approaches achieve a superior performance on a wide range of applications, e.g. face, dynamic texture and scene recognition.

We further find that for dynamic texture recognition, spatial information is more important than temporal information. We propose a “Super Histogram” to improve the reliability of LBP histogram. The proposed “Super Histogram” of PI-LBP and PD-LBP outperforms the state of the art on UCLA, DynTex and DynTex++ databases.

Chapter 7

A Chi-Squared-Transformed Subspace of LBP Histogram for Visual Recognition

Subspace approaches are often applied on the LBP histogram, whereas they are optimal under Gaussian assumption. We find that the LBP-histogram features deviate far from Gaussian distribution. To address this issue, we propose a Chi-squared transformation so that the resulting feature fits better to Gaussian distribution¹. Due to the asymmetric nature of our formulation, we propose a CST-APCA approach to better remove the unreliable dimensions in transformed feature space. Extensive experiments on face recognition, protein cellular classification, dynamic texture recognition show that the proposed feature transformation significantly enhances the performance gain of subspace approaches.

7.1 Introduction

LBP features has been widely used in many visual recognition tasks. However, the LBP histogram may not be reliably estimated due to: 1) The occurrence frequencies of LBP codes vary significantly. Histogram bins corresponding to infrequent patterns are poorly estimated. 2) To preserve locality, a small image patch is often used to construct the histogram, but it does not

¹This work is partially published in IEEE Transactions on Image Processing (journal paper [5] in author's publication list).

have sufficient pixels to produce a reliable histogram.

To remove the unreliable dimensions of LBP histogram or to derive a compact feature representation, subspace approaches are often applied. It is well-known that subspace approaches utilizing up to the second-order statistics are optimal under Gaussian assumption [12, 24, 25]. However, the LBP histogram has two properties that cause its underlying distribution deviates far from Gaussian. 1) Non-negative constraint: each histogram bin is non-negative. 2) Simplex constraint: all histogram bins sum to 1. As a result, the LBP histogram is bounded in the range of $[0, 1]$. In general, the expected value of LBP-histogram bin is close to 0. The lower bound at 0 has a severe truncation effect on the probability density function (PDF) of LBP histogram so that it largely deviates from Gaussian.

Our objective is to find a transformation that produces a feature with a distribution close to Gaussian. Towards this end, we propose a Chi-squared transformation based on the following observations. On one hand, the LBP-histogram bins are severely unbalanced, i.e. the occurrence frequencies of frequent patterns are much higher than those of infrequent patterns. On the other hand, the covariance matrix used in subspace approaches is constructed using Euclidean distance. Frequent patterns may overwhelm infrequent patterns when Euclidean distance is used. To enhance the discriminative power of infrequent patterns, distance measures other than Euclidean distance such as histogram intersection, Chi-squared distance and G-statistic are often employed [17, 26, 27]. The proposed transformation inherits the advantages of Chi-squared distance, which well handles the unbalanced nature of LBP histogram. More importantly, our analysis and experiments show that it produces a feature that fits better to Gaussian distribution.

The proposed Chi-squared transformation converts original classification problem into a two-class one. This problem is highly asymmetric since the number of inter-class sample pairs is much larger than the number of intra-class pairs. To handle this asymmetric nature, we propose to use asymmetric PCA [24, 25] instead of PCA to remove the unreliable dimensions in the CST feature space. We name the proposed approach as CST-APCA. It is validated extensively on

face recognition, protein cellular classification and dynamic texture recognition. The proposed CST-APCA consistently enhances the recognition performance in all experiments.

7.2 Related Work

Subspace approaches have been applied on LBP features to remove unreliable dimensions. For CENTRIST feature [3], principal component analysis (PCA) was applied on the LBP histogram of each patch. CENTRIST was improved in [82] by mining discriminative co-occurrence patterns. In [83], a random projection tree was utilized to encode LBP-like features, and PCA was applied to reduce the dimensionality. PCA has also been applied on the feature vector concatenated using LBP histograms of all patches [21, 22, 63, 84], and multi-level HOG-LBP feature [85]. For subspace approach, Mahalanobis distance is widely used in classification, in which the covariance matrix is built using Euclidean distance. However, due to the asymmetric nature of LBP histogram, those infrequent patterns may be easily overwhelmed by frequent patterns when Euclidean distance is used. Thus, other distance measures such as Chi-squared distance, histogram intersection and G-statistic [17, 26, 27] are often applied on the LBP histogram. The proposed approach inherits the advantages of Chi-squared distance and better handles the asymmetric nature of LBP histogram.

7.3 The Proposed Approach

7.3.1 Modeling the PDF of LBP Histogram

Due to its nonnegative and simplex constraints, the PDF of LBP histogram deviates far from Gaussian. The sample distributions of some LBP-histogram bins² of the AR face database [132]

²We use $LBP_{8,2}^{u2}$, a uniform LBP with 59 histogram bins extracted using 8 neighbors evenly spaced on a circle of radius 2.

Figure 7.1: Comparison of the distributions of the features before and after the proposed Chi-squared transformation. (a)-(d) Sample distributions of LBP-histogram features $h \in [0, 1]$. They look like a truncated Gaussian distribution. (e)-(h) The distributions of the transformed features $z \in [-1, 1]$, derived by applying the proposed transformation on the features with distributions shown in (a)-(d). They fit better to Gaussian distribution.

are plotted in Fig. 7.1(a)-(d). They look very much like a truncated Gaussian distribution, i.e. a Gaussian distribution truncated at the left-hand side. Thus, we use truncated Gaussian to model the PDF of LBP histogram. A truncated Gaussian random variable h has a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ within the interval $[a, b]$ and zero PDF elsewhere, as shown in Fig. 7.2. Its PDF is given by

$$f(h; \mu, \sigma, a, b) = \begin{cases} \frac{\frac{1}{\sigma} \phi(\frac{h-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})} & \text{if } h \in [a, b], \\ 0 & \text{otherwise.} \end{cases} \quad (7.1)$$

Here $\phi(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2)$ is the PDF of a standard Gaussian distribution, i.e. $x \sim \mathcal{N}(0, 1)$, and $\Phi(x)$ is its cumulative distribution function, i.e. $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-t^2/2} dt$. Denote $\alpha =$

Figure 7.2: Illustration of a truncated Gaussian distribution.

$\frac{a-\mu}{\sigma}, \beta = \frac{b-\mu}{\sigma}$. The mean and variance are:

$$\mu_T = \mu + \frac{\phi(\alpha) - \phi(\beta)}{\Phi(\beta) - \Phi(\alpha)}\sigma, \quad (7.2)$$

$$\sigma_T^2 = \sigma^2 \left[1 + \frac{\alpha\phi(\alpha) - \beta\phi(\beta)}{\Phi(\beta) - \Phi(\alpha)} - \left(\frac{\phi(\alpha) - \phi(\beta)}{\Phi(\beta) - \Phi(\alpha)} \right)^2 \right]. \quad (7.3)$$

Now we analyze the implications of nonnegative and simplex constraints of LBP histogram. Let $\mathbf{h} \in \mathbb{R}^d$ denote the LBP histogram, where $h_i \in [0, 1], i = 1, 2, \dots, d$ is one of its bins. The truncations happen at both sides, i.e. $a = 0, b = 1$. Since $\sum_{i=1}^d \mathbb{E}\{h_i\} = 1, \mathbb{E}\{h_i\} > 0$ for $\forall i$ and $d \gg 1$, $\mathbb{E}\{h_i\}$ is close to 0. Thus, the truncation at $a = 0$ is significant, whereas the truncation at $b = 1$ is almost negligible since it is at the far right tail. These are evidenced from Fig. 7.1(a)-(d). Thus, $\phi(\beta) \approx 0$ and $\Phi(\beta) \approx 1$. Then, we derive the simplified truncated Gaussian model for LBP histogram with mean and variance as:

$$\mu_S = \mu + \frac{\phi(\alpha)}{1 - \Phi(\alpha)}\sigma, \quad (7.4)$$

$$\sigma_S^2 = \sigma^2 \left[1 + \frac{\alpha\phi(\alpha)}{1 - \Phi(\alpha)} - \left(\frac{\phi(\alpha)}{1 - \Phi(\alpha)} \right)^2 \right]. \quad (7.5)$$

Now $\alpha = -\frac{\mu}{\sigma}$. We define Δ_μ and Δ_{σ^2} to measure how far the mean and variance of the truncated

Gaussian differ from non-truncated ones:

$$\Delta_\mu = \left| \frac{\mu_S - \mu}{\sigma} \right| = \frac{\phi(\alpha)}{1 - \Phi(\alpha)}, \quad (7.6)$$

$$\Delta_{\sigma^2} = \left| \frac{\sigma_S^2 - \sigma^2}{\sigma^2} \right| = \left| \frac{\alpha\phi(\alpha)}{1 - \Phi(\alpha)} - \left(\frac{\phi(\alpha)}{1 - \Phi(\alpha)} \right)^2 \right|. \quad (7.7)$$

It is not difficult to see that Δ_μ increases monotonically with α for $\alpha \leq 0$. We then study how Δ_{σ^2} varies with α by plotting it in Fig. 7.3. It shows that Δ_{σ^2} also increases monotonically with α . This is inline with our intuition. A smaller α indicates that the truncation at $a = 0$ happens at the far left tail, which will not alter the Gaussian distribution significantly.

Given mean μ_S and variance σ_S^2 of a truncated Gaussian, it is difficult to derive μ and σ of the untruncated Gaussian distribution. Thus, we define a truncation effect measure α_S directly using μ_S and σ_S ,

$$\alpha_S = \frac{a - \mu_S}{\sigma_S} = - \frac{-\alpha + \frac{\phi(\alpha)}{1 - \Phi(\alpha)}}{\sqrt{1 + \frac{\alpha\phi(\alpha)}{1 - \Phi(\alpha)} - \left(\frac{\phi(\alpha)}{1 - \Phi(\alpha)} \right)^2}}. \quad (7.8)$$

We then study how α_S varies with α by plotting it in Fig. 7.4. Clearly, α_S increases monotonically with α for $\alpha \leq 0$. Since $\alpha_S = f(\alpha)$ is a bijection function w.r.t. α for $\alpha \leq 0$, there is

Figure 7.3: Δ_{σ^2} increases monotonically with α .

Figure 7.4: α_S increases monotonically with α .

a unique inverse mapping from α_S to α , i.e. $\alpha = f^{-1}(\alpha_S)$, where $f^{-1}(\alpha_S)$ is a monotonically increasing function w.r.t. α_S . Therefore, Δ_μ and Δ_{σ^2} increase monotonically with α_S . A larger α_S , or equivalently a smaller $|\alpha_S|$ indicates a more severe truncation.

To study how severe the truncation is, we plot $|\alpha_S|$ for all LBP-histogram bins of the AR database in Fig. 7.5. For a better view, $|\alpha_S|$ is sorted in an ascending order. If $|\alpha_S| > 3$, it means that the truncation at $a = 0$ happens beyond three times of the standard derivation from the mean, and hence is negligible. However, from Fig. 7.5 we can see that for most LBP-histogram bins $|\alpha_S| < 2$ and $|\alpha_S|$ is clustered around 1, which indicates a severe truncation.

In summary, the PDF of the LBP-histogram feature deviates largely from Gaussian distribution, as shown in Fig. 7.1(a)-(d). We model it as a truncated Gaussian distribution, and show that $|\alpha_S| = \frac{|a - \mu_S|}{\sigma_S}$ can be used to measure how severe the truncation is. For most LBP-histogram bins $|\alpha_S| < 2$, and hence their PDFs are severely truncated. In the next section, we propose a novel transformation that greatly reduces the truncation effect, and hence the resulting feature fits much better to Gaussian distribution.

Figure 7.5: Plot of truncation measure $|\alpha_S|$ of the LBP-histogram features and $|\alpha_z|$ of the features after the proposed Chi-squared transformation. For a better view, $|\alpha_S|$ is sorted in an ascending order and plotted together with the corresponding $|\alpha_z|$. $|\alpha_z|$ is consistently much larger than $|\alpha_S|$, which indicates that the truncation is less severe after the proposed transformation.

7.3.2 Proposed Chi-Squared Transformation

LBP histogram has been used as a robust image representation for many recognition tasks. In real applications, it often produces a high-dimensional feature vector, which adversely affects the robust classification. It is hence not a surprise that many recent approaches utilize linear subspace methods to remove the unreliable dimensions in LBP histogram or to derive a compact feature representation [3, 21, 22, 63, 83–85]. However, subspace approaches utilizing up to the second-order statistics are only optimal when the underlying distribution is Gaussian [24, 25]. Under Gaussian assumption, the maximum a posterior (MAP) rule that minimizes the classification error is simplified to evaluating the following discriminant function [25]:

$$g_i(\mathbf{h}) = -\frac{1}{2}(\mathbf{h} - \bar{\mathbf{h}}_i)^T \Sigma_i^{-1}(\mathbf{h} - \bar{\mathbf{h}}_i) + \theta_i, \quad (7.9)$$

where $\bar{\mathbf{h}}_i$ is the class mean, Σ_i is the class covariance matrix and θ_i is a threshold to control the error rate of class i at a price of others. It is basically a minimum Mahalanobis distance classi-

fier (mMDC). As the covariance matrix Σ_i of individual class may not be reliably estimated in many applications, a pooled covariance matrix called within-class scatter matrix is often used instead in Eqn. (7.9). Unfortunately, the Mahalanobis distance cannot be reliably evaluated in the high-dimensional space due to limited training samples. Thus, various subspace approaches are developed to solve this problem. PCA+LDA [7] removes dimensions corresponding to unreliable small eigenvalues so that the Mahalanobis distance can be evaluated reliably in a small PCA+LDA subspace. Bayesian Maximum Likelihood approaches [10, 11] partition the feature space into two subspaces where the Mahalanobis distance is evaluated reliably in one subspace and the Euclidian distance is evaluated in the other subspace to circumvent the unreliable evaluation of the Mahalanobis distance. The Euclidean distance can be viewed as a degenerated Mahalanobis distance, where the covariance matrix is assumed to be a scalar matrix. ERE approach [12] regularizes the covariance matrix before computing the Mahalanobis distance to attenuate this problem.

However, it is shown in the previous section that the PDF of the LBP-histogram feature deviates far from Gaussian. Therefore, Mahalanobis distance and Euclidian distance are not optimal for the LBP histogram. This is evidenced by the fact that other distance measures such as Chi-squared distance, histogram intersection and G-statistic [17, 26, 27] have been widely used in the LBP feature space $\mathbb{R}^d$ and demonstrated a superior performance to Euclidian distance. (Mahalanobis distance cannot be applied in $\mathbb{R}^d$ because Σ_i is singular.) To utilize linear subspace approach in solving the high dimensionality problem of the LBP histogram, we propose to transform it to a new feature space so that the resulting feature fits better to Gaussian distribution.

Given two samples of the LBP-histogram feature represented by $\mathbf{x}, \mathbf{y}$, we define a new feature vector $\mathbf{z} = \{z_1, z_2, \dots, z_d\}$, where each element z_i is:

$$z_i = \frac{x_i - y_i}{\sqrt{x_i + y_i}}. \quad (7.10)$$

As the l_2 -norm of $\mathbf{z}$ is the Chi-squared distance between feature vectors $\mathbf{x}, \mathbf{y}$,

$$\mathbf{z}^T \mathbf{z} = \sum_{i=1}^d \frac{(x_i - y_i)^2}{x_i + y_i}, \quad (7.11)$$

we name the proposed transformation in Eqn. (7.10) as Chi-squared transformation.

We have shown that for most LBP histogram bins, $|\alpha_S| < 2$, and hence their PDFs are severely truncated. Now we show that after the proposed Chi-squared transformation, the resulting feature has much less truncation effect. Suppose we have a LBP-histogram bin h that has a truncated Gaussian distribution with mean μ_S , and two independent samples x, y drawn from this distribution. We first calculate the mean $\mu_z = \mathbb{E}_h\{\frac{x-y}{\sqrt{x+y}}\}$. For every sample pair $\{x, y\}$, there is a corresponding sample pair $\{y, x\}$, i.e. $\tilde{z} = \frac{y-x}{\sqrt{x+y}}$. The summation of z and $\tilde{z}$ is zero, thus $\mu_z = 0$. We then calculate the variance $\sigma_z^2 = \mathbb{E}_h\{\frac{(x-y)^2}{x+y}\} = \mathbb{E}_h\{x + y - \frac{4xy}{x+y}\} \leq \mathbb{E}_h\{x+y\} = 2\mu_S$. Thus, σ_z^2 is upper-bounded by $2\mu_S$. As $x, y \in [0, 1]$, the PDF of z is truncated in $[-1, 1]$. Similar to Eqn. (7.8), the following criterion is defined for the resulting feature z to measure how severe the truncation is:

$$|\alpha_z| = \frac{|a_z - \mu_z|}{\sigma_z} = \frac{1}{\sigma_z}. \quad (7.12)$$

Since $\sigma_z \leq \sqrt{2\mu_S}$, $|\alpha_z| \geq \frac{1}{\sqrt{2\mu_S}}$. We have shown that for LBP-histogram bin, $\mu_S \ll 1$. It is easy to satisfy $\frac{1}{\sqrt{2\mu_S}} > 3$. Thus, $|\alpha_z| > 3$, which means that the PDF of the feature after the proposed transformation is truncated beyond three times of the standard derivations from the mean, and hence the truncation is almost negligible. We apply the proposed transformation on the LBP-histogram features of the AR database, and plot $|\alpha_z|$ for the resulting features in Fig. 7.5. $|\alpha_z|$ is much larger than $|\alpha_S|$. After the proposed transformation, the truncation effect is greatly reduced.

Finally, we visually compare the distributions of the features before and after the proposed transformation. We apply the proposed CST on the LBP-histogram features with distributions

Figure 7.6: Illustration of the skewness and the kurtosis.

shown in Fig. 7.1(a)-(d), and plot the histograms of the resulting features in Fig. 7.1(e)-(h), respectively. After the proposed transformation, the truncations happen at the far tails. Clearly, the PDF of the resulting feature has much less truncation effect and visually fits better to Gaussian. In the next section, we will further analyze the normality using moments.

7.3.3 Normality Analysis Using Moments

Moments have been widely used to test the normality of a distribution [169–172]. Among them, skewness and kurtosis are most simple but effective measures. As shown in Fig. 7.6, the skewness is a measure of the asymmetry of a PDF about its mean, and the kurtosis is a measure of the “peakedness” of a distribution, which are respectively defined as:

$$s = m_3/m_2^{3/2}, \quad (7.13)$$

$$k = m_4/m_2^2, \quad (7.14)$$

where $m_r = \mathbb{E}_h\{(h - \mu_h)^r\}$ is the r -th order moment, and μ_h is the mean. For a Gaussian distribution, $s = 0$ and $k = 3$. Deviation from these two values indicates non-normality.

7.3.3.1 Skewness and Kurtosis of LBP Histogram

Based on the results from [173], the r -th moment $m_r(\alpha, \beta)$ for the standard Gaussian distribution $\mathcal{N}(0, 1)$ truncated in the range of (α, β) is calculated as:

$$m_{2k}(\alpha, \beta) = (2k-1)!! \left(1 - \sum_{i=1}^k \frac{1}{(2i-1)!!} \frac{[z^{2i-1}\phi(z)]_\beta^\alpha}{[\Phi(z)]_\beta^\alpha} \right) \quad \text{for } k = 1, 2, \dots,$$

$$m_{2k+1}(\alpha, \beta) = - \sum_{i=0}^k \frac{(2k)!!}{(2i)!!} \frac{[z^{2i}\phi(z)]_\beta^\alpha}{[\Phi(z)]_\beta^\alpha} \quad \text{for } k = 0, 1, \dots,$$

where $n!!$ is the double factorial, i.e. $n!! = n \times (n-2)!!$ if $n \geq 2$, and $n!! = 1$ if $n = -1, 0$ or 1 . $[f(z)]_\beta^\alpha = f(\alpha) - f(\beta)$. As $(\alpha, \beta) \rightarrow (-\inf, \inf)$, $m_{2k}(\alpha, \beta)$ and $m_{2k+1}(\alpha, \beta)$ tend to classical Gaussian moments $(2k-1)!!$ and 0 , respectively.

We have shown that the PDF of the LBP-histogram feature is severely truncated at the left tail, but the truncation at the right tail is almost negligible, i.e. $\phi(\beta) \approx 0$, $\Phi(\beta) \approx 1$. After plugging them into $m_{2k}(\alpha, \beta)$ and $m_{2k+1}(\alpha, \beta)$, the first four moments can be obtained. After some algebra, the skewness and the kurtosis of the LBP-histogram feature under truncated Gaussian model are derived as:

$$s_t = m_3/m_2^{3/2} = \frac{\frac{2\phi(\alpha)+\alpha^2\phi(\alpha)}{1-\Phi(\alpha)}}{\left(1 + \frac{\alpha\phi(\alpha)}{1-\Phi(\alpha)}\right)^{3/2}}, \quad (7.15)$$

$$k_t = m_4/m_2^2 = \frac{3\left(1 + \frac{\alpha\phi(\alpha)+\frac{1}{3}\alpha^3\phi(\alpha)}{1-\Phi(\alpha)}\right)}{\left(1 + \frac{\alpha\phi(\alpha)}{1-\Phi(\alpha)}\right)^2}. \quad (7.16)$$

Take note that $\alpha = (a - \mu)/\sigma = -\mu/\sigma$.

7.3.3.2 Skewness and Kurtosis of z After CST

Previously, it is shown that $\mu_z = 0$. The p -th moment of z is calculated as: $\mathbb{E}_z\{z^p\} = \mathbb{E}_h\{(\frac{x-y}{\sqrt{x+y}})^p\}$. For every sample pair $\{x, y\}$, we have a corresponding sample pair $\{y, x\}$, i.e. $\tilde{z} = \frac{y-x}{\sqrt{x+y}}$, such that $z^p + \tilde{z}^p = 0$ for odd p . Thus, the odd moment of z is zero, and hence the skewness $s_z = 0$.

We now derive the kurtosis of z . Previously, $m_k(\alpha, \beta)$ is derived for a standard Gaussian distribution $\mathcal{N}(0, 1)$ truncated in (α, β) . If the Gaussian distribution before truncation is $\mathcal{N}(\mu, \sigma)$, $m_k(\alpha, \beta) = \mathbb{E}_h\{(\frac{h-\mu}{\sigma})^k\}$. Then, the raw moment $m_{k,u} = \mathbb{E}_h\{h^k\}$ for the LBP-histogram feature can be calculated in an iterative manner, i.e.

$$m_{k,u} = \sigma^k m_k(\alpha, \beta) + \sum_{i=1}^k \frac{k!(-1)^{i+1} \mu^i m_{k-i,u}}{i!(k-i)!}. \quad (7.17)$$

The second-order moment of feature z is $m_{2,z} = \mathbb{E}_h\{\frac{(x-y)^2}{x+y}\}$. We apply the Taylor expansion up to the second order for $\frac{1}{x+y}$ around $\mathbb{E}_h\{x+y\}$,

$$\begin{aligned} m_{2,z} = \mathbb{E}_h\{ & \frac{(x-y)^2}{\mathbb{E}_h\{x+y\}} - \frac{(x-y)^2(x+y - \mathbb{E}_h\{x+y\})}{\mathbb{E}_h^2\{x+y\}} \\ & + \frac{(x-y)^2(x+y - \mathbb{E}_h\{x+y\})^2}{\mathbb{E}_h^3\{x+y\}} \}. \end{aligned}$$

Our analysis shows that $m_{2,z}$ is well approximated using the first term only. By including other two terms, it only slightly improves the approximation accuracy, but greatly increases the computational complexity. For simplicity, $m_{2,z}$ is approximated by $\mathbb{E}_h\{\frac{(x-y)^2}{\mathbb{E}_h\{x+y\}}\}$. Thus, $m_{2,z} \approx \frac{\mathbb{E}_h\{(x-y)^2\}}{\mathbb{E}_h\{x+y\}} = \frac{m_{2,u} - m_{1,u}^2}{m_{1,u}}$. Similarly, $m_{4,z} \approx \frac{\mathbb{E}_h\{(x-y)^4\}}{\mathbb{E}_h\{(x+y)^2\}}$. After some algebra, we obtain $\mathbb{E}_h\{(x-y)^4\} = 2m_{4,u} - 8m_{3,u}m_{1,u} + 6m_{2,u}^2$ and $\mathbb{E}_h\{(x+y)^2\} = 2m_{2,u} + 2m_{1,u}^2$. k_z is then approximated by

$$k_z \approx \frac{m_{4,u} - 4m_{3,u}m_{1,u} + 3m_{2,u}^2}{m_{2,u} + m_{1,u}^2} \times \frac{m_{1,u}^2}{(m_{2,u} - m_{1,u}^2)^2} \quad (7.18)$$

Figure 7.7: Comparison of the Jarque-Bera test statistic J_t before and J_z after the proposed Chi-squared transformation.

7.3.3.3 Jarque-Bera test

We analyze the normality of the features before and after the proposed CST using Jarque-Bera test [169], which utilizes both skewness s and kurtosis k :

$$J = \frac{n}{6} \left(s^2 + \frac{(k-3)^2}{4} \right), \quad (7.19)$$

where n is the number of samples. For a Gaussian distribution, $s = 0$ and $k = 3$. Thus, a smaller J indicates a better fit to Gaussian distribution.

We compare the Jarque-Bera test statistic J_t before and J_z after the proposed Chi-squared transformation. Both J_t and J_z can be represented as a function of α that indicates the truncation effect at $a = 0$. We plot J_t and J_z vs. α in Fig. 7.7. J_z is consistently smaller than J_t . Thus, we conclude that the proposed CST produces a feature that fits better to Gaussian distribution under the Jarque-Bera test.

7.3.4 Chi-Squared-Transformed Subspace Analysis

The transformed feature $\mathbf{z}$ is defined on a pairwise basis, i.e. each element $z_i = \frac{x_i - y_i}{\sqrt{x_i + y_i}}$ is defined using a sample pair $\{\mathbf{x}, \mathbf{y}\}$. It naturally transfers the original classification problem into a two-class one: class Ω_I contains the sample differences from the same class, and class Ω_E contains the sample differences between different classes. It is easy to show that both classes have zero mean. The covariance matrices for class Ω_I and Ω_E are respectively constructed as $\Sigma_I = \frac{1}{N_I} \sum_{\mathbf{z} \in \Omega_I} \mathbf{z} \mathbf{z}^T$ and $\Sigma_E = \frac{1}{N_E} \sum_{\mathbf{z} \in \Omega_E} \mathbf{z} \mathbf{z}^T$, where N_I and N_E are the number of samples for class Ω_I and Ω_E , respectively.

Eqn. (7.9) that minimizes the classification error decides that a sample $\mathbf{z}$ belongs to class Ω_I if

$$\mathbf{z} \Sigma_E^{-1} \mathbf{z} - \mathbf{z} \Sigma_I^{-1} \mathbf{z} > \theta, \quad (7.20)$$

where θ is a threshold. Let λ_k^I, λ_k^E and Φ_k^I, Φ_k^E denote the eigenvalues and eigenvectors of Σ_I, Σ_E , respectively. Let p_k and q_k denote the projection of $\mathbf{z}$ on Φ_k^I and Φ_k^E , respectively. The decision rule is simplified to

$$\sum_{k=1}^d \frac{p_k^2}{\lambda_k^E} - \sum_{k=1}^d \frac{q_k^2}{\lambda_k^I} > \theta. \quad (7.21)$$

Eigenvalues λ_k^E and λ_k^I are an estimate of the class true (ensemble) variance based on available training data. The decision rule will overfit the training samples, and hence result in a poor generalization performance if these eigenvalues differ significantly from the ensemble variances.

As pointed out in [12, 24, 25], in general the large eigenvalues are biased upward and the small ones are biased downward. The bias is most significant when the eigenvalues are equal, and less severe when the eigenvalues are highly disparate. Furthermore, the eigenvalues often first decay very rapidly and then stabilize. Thus, the small eigenvalues are biased much more

severely than the large ones. Since the inverses of eigenvalues are used to weight the features, these small eigenvalues are even more problematic.

PCA applies eigen-decomposition on the total scatter matrix $\Sigma_t = \Phi_t \Lambda_t \Phi_t^T$, where Σ_t is calculated by

$$\Sigma_t = \frac{N_I}{N_I + N_E} \Sigma_I + \frac{N_E}{N_I + N_E} \Sigma_E. \quad (7.22)$$

Here, the between-class scatter matrix is zero. PCA keeps m eigenvectors Φ_m corresponding to the largest eigenvalues. By removing the subspace spanned by eigenvectors Φ_k^E and Φ_k^I corresponding to small eigenvalues, it improves the generalization performance on the novel testing data. Other subspace approaches improve the performance explicitly or implicitly in a similar way.

PCA is designed to achieve the least-mean-square reconstruction error. However, for optimal classification, the goal is to remove the dimensions in which the sample-based class-conditional variances are unreliable. In our case, $N_E \gg N_I$. More training samples of a class may result in a more reliable covariance matrix if they are properly collected. Thus, Σ_E is more reliable than Σ_I . A larger weight rather than $N_I/(N_I + N_E)$ should be assigned to unreliable Σ_I in Eqn. (7.22) so that more dimensions characterized by the small variances of this class can be removed. On the other hand, Σ_I represents the intrinsic class variations, whereas Σ_E represents a universal background model. Compared with Σ_I , the eigenspectrum of Σ_E tends to be flatter. A flatter eigenspectrum may deviate more from the ensemble variances. Thus, a larger weight should be assigned to Σ_E . In summary, PCA on Σ_t does not effectively remove the unreliable dimensions. To tackle this problem, we construct an asymmetric pooled covariance matrix by

$$\Sigma_\gamma = \gamma \Sigma_I + (1 - \gamma) \Sigma_E, \quad (7.23)$$

where γ is the weighting factor of Σ_I and $0 \leq \gamma \leq 1$.

If $\gamma = N_I/(N_E + N_I)$, Σ_γ is equivalent to Σ_t . Without any prior knowledge, we assign the weight proportional to the inverse of the number of samples since less training samples will result in a less reliable covariance matrix, i.e. $\gamma = N_E/(N_E + N_I)$. On the other hand, the eigenspectrum of Σ_E is flatter and hence less reliable. A larger weight is needed for Σ_E . Hence, the optimal γ should be in the range of $N_I/(N_E + N_I) < \gamma < N_E/(N_E + N_I)$.

To remove the dimensions corresponding to the small eigenvalues of Σ_I and Σ_E that are harmful to reliable classification, eigen-decomposition is applied on Σ_γ ,

$$\Sigma_\gamma = \Phi_\gamma \Lambda_\gamma \Phi_\gamma^T, \quad (7.24)$$

where Φ_γ and Λ_γ correspond to the eigenvectors and eigenvalues of Σ_γ , respectively. We select eigenvectors $\Phi_{\gamma,m}$ corresponding to m largest eigenvalues, and project $\mathbf{z}$ into the subspace as:

$$\hat{\mathbf{z}} = \Phi_{\gamma,m}^T \mathbf{z}. \quad (7.25)$$

After removing the unreliable dimensions corresponding to small eigenvalues that are biased downwards, the bias problem of eigenspectrum is not fully solved, i.e. the large eigenvalues are still biased upwards. As discussed earlier, the eigenspectrum of Σ_E is estimated more reliably using much more samples. Thus, the remaining large eigenvalues of Σ_I are biased upwards more severely than Σ_E . To correct this, we use the following classifier:

$$\hat{\mathbf{z}} \hat{\Sigma}_E^{-1} \hat{\mathbf{z}} - \hat{\mathbf{z}} (\psi \hat{\Sigma}_I)^{-1} \hat{\mathbf{z}} > \hat{\theta}, \quad (7.26)$$

where $\hat{\Sigma}_E = \Phi_{\gamma,m}^T \Sigma_E \Phi_{\gamma,m}$, $\hat{\Sigma}_I = \Phi_{\gamma,m}^T \Sigma_I \Phi_{\gamma,m}$, and ψ is a weighting constant. The value of ψ depends on how far the large eigenvalues of Σ_I deviate from their ensemble variances compared with those of Σ_E .

We name the proposed approach as CST-APCA, i.e. first apply the proposed Chi-squared

transformation on the LBP histogram to derive a feature that fits better to Gaussian distribution, and then apply asymmetric PCA on the resulting feature to remove unreliable dimensions. In the next section, we evaluate the proposed approach on various recognition tasks.

7.4 Experimental Results

To validate the proposed Chi-squared transformation, we compare the features before and after the proposed transformation under the normality test, and conduct comparison experiments on face recognition, protein cellular classification and dynamic texture classification.

The nearest-neighbor classifier (NNC) with Euclidean distance in the original LBP feature space is used as the baseline algorithm, denoted as NNC-ED. As Euclidean distance may not well handle the infrequent patterns of LBP histogram, other distance measures such as Chi-squared distance are often used [17, 26, 27]. Hence, the second compared algorithm is NNC + Chi-squared distance, denoted as NNC-Chi2D. As the dimensionality of LBP feature is often high, Mahalanobis distance cannot be directly calculated in the original LBP feature space, and PCA is often applied to reduce the dimensionality [3, 21, 22, 63, 83–85]. Thus, the third compared algorithm is PCA-mMDC.

We also compare the proposed algorithm with other state-of-the-art approaches. Power mean support vector machine (PmSVM) [174] has achieved a superior performance in classifying LBP features. It is claimed in [174] that when features are histograms, SVMs of additive kernels are most effective. PmSVM utilizes a special family of additive kernels, i.e.

$$M_p(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^d \left(\frac{x_i^p + y_i^p}{2} \right)^{1/p}, \quad (7.27)$$

for any $p < 0$. It includes some popular additive kernels, e.g. χ^2 kernel if $p = -1$ and histogram-intersection kernel if $p = -\infty$. We compare the proposed approach with both, i.e.

PmSVM-Chi2 and PmSVM-HI.³ As suggested in [174], we use $p = -8$ for PmSVM-HI, and cost parameter $C = 0.01$ in all experiments. Very recently, a simple transform that square-roots every feature value [175] is shown to improve the fitness to Gaussian for video features. We use it with PCA-mMDC, and name it as SR-PCA-mMDC. Besides these, the proposed approach is also compared with other state-of-the-art solutions to dynamic texture recognition.

The proposed transformation can be also applied to other histogram features such as HOG features [15] to improve its fitness to Gaussian. To verify this, we conduct a comparison experiment on the LFW dataset using HOG features.

To illustrate the advantages of the proposed transformation, we compare it with those proposed approaches in previous chapters, i.e. NRLBP and ENRLBP in Chapter 4, LBP structure optimization with MmCMI/MCMI in Chapter 5, and PI-LBP/PD-LBP in Chapter 6.

7.4.1 Jarque-Bera Normality Test

In our previous analysis, we have shown that the proposed Chi-squared transformation results in a feature that fits better to Gaussian distribution. Now, we conduct Jarque-Bera test [169] on real images: the AR face database, the O2FN face database and the 2D Hela database. More information on these datasets can be found in the latter sections. The results on the UCLA dataset and the Dyntex++ dataset are similar to these three, and hence omitted. We plot J_t and J_z in Fig. 7.8, which represent the test statistics for features before and after the proposed Chi-squared transformation, respectively. A smaller test statistic indicates a better fit to Gaussian distribution. For a better view, J_t is sorted in an ascending order, and plotted together with the corresponding J_z .

We can see that J_t deviates far from 0, which indicates that the LBP-histogram feature deviates far from Gaussian distribution. After the proposed Chi-squared transformation, on all three

³We download the source codes for power mean SVM from <https://sites.google.com/site/wujx2001/home/power-mean-svm>.

databases J_z is consistently and significantly smaller than J_t . This validates our previous analysis that the proposed Chi-squared transformation produces a feature that fits better to Gaussian distribution under Jarque-Bera test.

7.4.2 Face Recognition on the AR Database

We normalize the image to 150×130 pixels, apply an ellipse mask to remove background pixels and divide it into patches of 21×18 pixels. The feature descriptor is $LBP_{8,2}^{u2}$. Initial experiments show that a better performance is achieved when patches are overlapped. The best one is reported, i.e. 50% overlap between neighboring patches. In total, there are $13 \times 13 = 169$ patches. The concatenated feature has $59 \times 169 = 9971$ dimensions. 75 subjects are chosen from the AR database [132], each with two sections. Each section contains 7 images: one neutral, 3 with different facial expressions and 3 in different illumination conditions. We use the first section for training and the second for testing.

The comparison with other approaches is shown in Table 7.1. NNC-Chi2D performs better than NNC-ED, which indicates that Chi-squared distance is a better measure for LBP features. PCA-mMDC performs even worse than NNC-Chi2D, mainly because of non-normality of LBP

Figure 7.8: Comparison of the test statistic J_t and J_z for features before and after the proposed Chi-squared transformation on (a) the AR database, (b) the O2FN database, (c) the 2D Hela database. For a better view, J_t is sorted in an ascending order. On all three databases, J_z is consistently and significantly smaller than J_t .

Table 7.1: Comparison of error rates for different approaches on the AR database.

Algorithms	Error Rate
NNC-ED	3.43%
NNC-Chi2D	1.33%
PCA-mMDC	2.48%
SR-PCA-mMDC [175]	1.33%
PmSVM-Chi2 [174]	0.00%
PmSVM-HI [174]	0.00%
Proposed NRLBP	0.95%
Proposed ENRLBP	1.14%
Proposed PI-LBP	1.52%
Proposed PD-LBP	0.95%
Proposed CST-APCA	0.38%

features. SR-PCA-mMDC outperforms PCA-mMDC, as it improves the fitness to Gaussian by square-rooting features. The proposed CST-APCA performs much better than PCA-mMDC, which indeed demonstrates the superiority of the proposed transformation. NRLBP and ENRLBP do not show much advantages over other approaches, as the AR database has very little noise. PI-LBP and PD-LBP slightly outperform PCA-mMDC. After handling the non-Gaussian distribution of LBP features, the proposed CST-APCA significantly reduces the error rate to 0.38%. Surprisingly, PmSVM-Chi2 and PmSVM-HI achieve an error-free classification on this dataset.

7.4.3 Face Recognition on the O2FN Mobile Database

The O2FN mobile database [136] contains 50 subjects, each with 40 face images of size 144×176 pixels. The images are captured using a front camera of a mobile phone with native settings, and hence they are of low image quality and severely distorted by noise. The images also contain some pose variations and significant illumination variations. We use first 20 images of each subject for training and the rest for testing. LBP features are extracted similarly as for the AR database, which have 9971 dimensions.

The comparison with others is shown in Table 7.2. NNC-Chi2D slightly reduces the error rate compared with NNC-ED. As images of the O2FN database are noisy, many dimensions of LBP features are not reliable. PCA-mMDC greatly reduces the classification error by removing these unreliable dimensions. SR-PCA-mMDC further boosts the performance by improving the fitness to Gaussian. NRLBP and ENRLBP further improves the performance by recovering the corrupted LBP codes. On top of these, the proposed CST-APCA outperforms all compared approaches.

Table 7.2: Comparison of error rates for different approaches on the O2FN database.

Algorithms	Error Rate
NNC-ED	33.20%
NNC-Chi2D	30.00%
PCA-mMDC	5.00%
SR-PCA-mMDC [175]	2.80%
PmSVM-Chi2 [174]	4.00%
PmSVM-HI [174]	4.10%
Proposed NRLBP	<u>1.90%</u>
Proposed ENRLBP	<u>1.80%</u>
Proposed CST-APCA	<u>1.50%</u>

7.4.4 Face Recognition on the LFW Database

The Labeled Face in the Wild (LFW) database has been widely used as a benchmark dataset to evaluate face recognition algorithm [63, 176, 177]. It consists of 13,233 images of 5,749 subjects. All images are collected from the internet with large pose, illumination and facial-expression variations, as well as occlusions.

In [63], it is shown that high-dimensional LBP features achieve a better performance than baseline LBP features. In this chapter, we report the results on both datasets.⁴ In image understanding and recognition, histogram-like features are widely used, e.g. LBP histogram,

⁴Features can be downloaded from <http://home.ustc.edu.cn/~chendong/>.

histogram of oriented gradient (HOG), scale-invariant feature transform, color histogram and gray-level co-occurrence histogram. The proposed transformation can improve the performance of other histogram-like features. To verify this, we conduct experiments on HOG features. These features are extracted as follow:

Baseline LBP Feature: The image is normalized to 100×100 pixels by an affine transformation derived using 5 landmarks (two eyes, nose and two mouth tips). Then, it is partitioned into 10×10 non-overlapped patches. An LBP histogram is extracted from each patch. All histograms are concatenated to form the final feature vector of $59 \times 10^2 = 5900$ dimensions.

Baseline HOG Feature: The image is normalized to 128×128 pixels. A sliding window of 16×16 pixels is used to scan the whole image, at the stripe size of 8 pixels. Each sliding window is divided into 4 cells. A HOG histogram of 9 bins is extracted from each cell. The histograms of one window are l_2 -normalized. The concatenated feature vector has $9 \times 4 \times 15^2 = 8100$ dimensions.

High-Dim LBP Feature: The normalized images are resized to five scales, i.e. the side lengths in each scale are 300, 212, 150, 106 and 75. In each scale, an image cell of 40×40 pixels is extracted from each of 27 facial landmarks, and further divided into non-overlapped patches of 10×10 pixels. An LBP histogram is extracted from each patch. The concatenated feature vector has $59 \times 27 \times 5 \times 16 = 127,440$ dimensions. This high-dim LBP feature has achieved the state-of-the-art performance on the LFW dataset [63], but also imposes a great challenge on both memory and computational complexity.

We report the averaged results over 10 folds of View 2. More precisely, 3000 positive pairs (two images of the same subject) and 3000 negative pairs (two images of different subjects) are divided into 10 subsets. In each fold, 9 subsets are used as the training set and the hold-out subset is used for testing. The task is to determine whether an image pair comes from the same subject or not. We follow the LFW’s restricted protocol. No outside dataset is used for training.

We test several subspace approaches with the proposed transformation. ERE approach [12] performs best, and hence we report this result. The results are summarized in Table 7.3. The results of Joint Bayesian [63] are copied here for comparison.

Table 7.3: Comparison of error rates for different approaches on the LFW database.

Algorithms	Error Rate		
	Baseline LBP	Baseline HOG	High-Dim LBP
NNC-ED	32.70%	35.95%	28.70%
NNC-Chi2D	28.70%	34.12%	26.35%
PCA-mMDC	25.92%	30.27%	21.05%
SR-PCA-mMDC [175]	19.43%	27.62%	15.88%
PmSVM-Chi2 [174]	21.17%	27.02%	13.67%
PmSVM-HI [174]	21.20%	26.88%	13.43%
Joint Bayesian [63]	19.95%	-	15.92%
Proposed CST-APCA	<u>15.42%</u>	<u>20.00%</u>	<u>9.50%</u>

For all approaches, baseline LBP performs poor than high-dim LBP, but better than baseline HOG. NNC-Chi2D consistently outperforms NNC-ED. PCA-mMDC marginally reduces the error rate compared with these two. SR-PCA-mMDC further reduces the error, and achieves even slightly better performance than joint Bayesian approach [63]. PmSVMs perform worse than SR-PCA-mMDC on baseline LBP, but better on baseline HOG and high-dim LBP. It shows that PmSVMs are more effective on higher-dimensional features. The proposed approach further reduces the error rates on all three datasets. Especially on high-dim LBP feature, it significantly decreases the error rate of joint Bayesian [63], the published best results, from 15.92% to 9.50%.

7.4.5 Protein Cellular Classification on the 2D Hela Database

Protein cellular classification is useful for characterizing newly discovered genes. The 2D Hela database [138] contains 862 single-cell images (16-bit gray image of size 382×382 pixels). It has 10 classes and each with more than 70 images. Some samples are shown in Fig. 7.9.

Figure 7.9: Sample images of the 2D Hela database. Images (a)-(j) correspond to actin, DNA, endosome, ER, golgia, golgpp, lysosome, microtubules, mitochondria and nucleolus, respectively.

We extract LBP features at multiple scales, i.e. $LBP_{8,1}$, $LBP_{8,2}$ and $LBP_{8,3}$, same as in [140]. At each scale, the histogram has $2^8 = 256$ bins. Protein cellular only occupies a small portion of the image. The background pixels are harmful for reliable classification, and mainly fall into bin 256. Thus, bin 256 at each scale is removed. These histograms are concatenated to form the final feature vector of 765 dimensions. We randomly choose 80% of the images for training and the rest 20% for testing. The experiments are repeated five times and the average performance is reported in Table 7.4.

Table 7.4: Comparison of error rates for different approaches on the 2D Hela database.

Algorithms	Error Rate
NNC-ED	30.81%
NNC-Chi2D	24.53%
PCA-mMDC	13.49%
SR-PCA-mMDC [175]	9.42%
PmSVM-Chi2 [174]	11.63%
PmSVM-HI [174]	12.91%
Proposed CST-APCA	8.60%

As shown in Table 7.4, NNC-Chi2D significantly outperforms NNC-ED in the LBP feature space. PCA-mMDC greatly reduces the error rate, which shows the effectiveness of subspace

approaches in removing the unreliable dimensions of LBP features. SR-PCA-mMDC further boosts the performance compared with PCA-mMDC by improving the fitness to Gaussian. On top of these, the proposed CST-APCA further reduces the error rate to 8.60%. The transformed features fit better to Gaussian distribution, and hence yield a better performance.

7.4.6 Dynamic Texture Recognition on the UCLA Database

Dynamic texture (DT) is a sequence of images of moving scenes that exhibit certain stationarity properties in time [5, 150]. The UCLA database [156, 157] has been widely used as a benchmark dataset for DT recognition [5, 23, 158, 178]. It has 50 classes, each with 4 sequences. Several breakdowns were used to evaluate the dataset [5]. Among them, we choose 3 representative settings:

50-Class: Original setting. 4-fold cross-validation is used.

9-Class: Sequences of different viewpoints are grouped as 9 classes to evaluate DT recognition under viewpoint changes. We use the same settings as in [5, 23]. We randomly choose half of the dataset for training and the other half for testing. The average recognition rate over 10 trials is reported.

Shift-invariant Recognition (SIR): Each sequence is spatially partitioned into non-overlapping left and right halves. “Shift-invariant recognition” [178] was implemented to compare sequences only between halves. This setting is very challenging. We report both rank-1 (R1) and rank-5 (R5) recognition rates.

We use LBP-TOP [19] as the feature descriptor, in which LBP features are extracted from three orthogonal planes, *i.e.* XY-plane (spatial LBP), XT-plane and YT-plane, and concatenated to form the final feature vector. We try different ways to partition DTs. The best performance is achieved when we divide DTs along time axis into two patches for 50-class and SIR, and spatially divide sequences into 3×3 patches for 9-class. The feature dimensionality is 1536,

6912, 1536 for 50-class, 9-class and SIR, respectively. The results are shown in Table 7.5.

Table 7.5: Comparison with the state of the arts on the UCLA DT dataset for 50-class, 9-class and SIR settings.

Method	Error Rate			
	50-Class	9-Class	SIR-R1	SIR-R5
Distributed spacetime orientation [178]	19.0%	-	57.7%	40.0%
DL-PEGASOS [23]	1.0%	4.4%	-	-
DFS [5]	0.0%	2.5%	-	26.2%
NNC-ED	10.5%	2.3%	22.0%	7.7%
NNC-Chi2D	10.0%	2.2%	14.0%	5.0%
PCA-mMDC	0.5%	2.1%	17.7%	4.0%
SR-PCA-mMDC [175]	0.5%	1.5%	16.5%	3.8%
PmSVM-Chi2 [174]	7.5%	2.9%	13.5%	-
PmSVM-HI [174]	7.5%	2.7%	13.5%	-
Proposed approach with MmCMI	0.5%	1.6%	16.7%	7.7%
Proposed approach with MCMI	0.0%	1.6%	14.2%	5.5%
Proposed PI-LBP/PD-LBP	0.0%	1.8%	39.2%	26.2%
Proposed CST-APCA	<u>0.0%</u>	<u>1.0%</u>	<u>12.7%</u>	<u>2.2%</u>

On the 50-class setting, the proposed CST-APCA achieves an error-free classification. On the 9-class setting, previous published lowest error rate is 2.5% achieved by DFS [5]. PCA-mMDC slightly improves the performance, and SR-PCA-mMDC boosts it further. Proposed approach with MmCMI/MCMI and PI-LBP/PDLBP achieves a comparable performance to SR-PCA-mMDC. On top of these, the proposed approach significantly reduces it to 1.0%. On the most challenging SIR setting, the published best result is R5 error rate of 26.2% achieved by DFS [5]. Proposed PI-LBP/PD-LBP achieve a comparable performance to DFS. By utilizing a larger spatial-temporal neighborhood and optimizing the LBP structure, proposed approach with MmCMI/MCMI significantly reduces the error rate. By removing unreliable dimensions, PCA-mMDC and SR-PCA-mMDC further decrease the error rate, and the proposed CST-APCA further significantly decreases it to 2.2%. On the SIR-R1 setting, the proposed CST-APCA also significantly reduces the error rate compared with PCA-mMDC and SR-PCA-mMDC. In

summary, on the UCLA dataset, the proposed CST-APCA achieves an error-free classification on the 50-class setting, and a better performance on the 9-class and SIR settings compared with the state-of-the-art approaches.

7.4.7 Dynamic Texture Recognition on the DynTex++ Dataset

The DynTex++ dataset proposed in [23] aims to provide a rich benchmark for DT recognition. It has 36 classes, each with 100 sequences of size $50 \times 50 \times 50$. LBP-TOP [19] is used as the feature descriptor, which has 768 dimensions. We use the same setup as in [5, 23]. For each class, 50 sequences are randomly chosen as the training set, and the other 50 are used in testing. The experiments are repeated 5 times and the average performance is reported in Table 7.6.

Table 7.6: Comparison with the state of the arts on the DynTex++ DT dataset.

Algorithms	Error Rate
DL-PEGASOS [23]	36.30%
DFS [5]	10.10%
NNC-ED	10.43%
NNC-Chi2D	5.92%
PCA-mMDC	5.48%
SR-PCA-mMDC [175]	4.60%
PmSVM-Chi2 [174]	11.79%
PmSVM-HI [174]	11.83%
Proposed approach with MmCMI	4.30%
Proposed approach with MCMI	4.10%
Proposed PI-LBP/PD-LBP + NNC	8.10%
Proposed PI-LBP/PD-LBP + RBF SVM	7.60%
Proposed CST-APCA	<u>3.02%</u>

In literature, the lowest error rate reported on the DynTex++ dataset was 10.10% achieved by DFS [5]. Compared with DFS, proposed PI-LBP/PD-LBP significantly reduces the error rate. When NNC-Chi2D is used with LBP-TOP, the error rate is further reduced. PCA-mMDC on LBP-TOP significantly outperforms NNC-ED, but only marginally outperforms NNC-Chi2D. SR-PCA-mMDC slightly reduces the error rate compared with PCA-mMDC. LBP Structure op-

timization with MmCMI/MCMI achieves a slightly better performance than SR-PCA-mMDC. The proposed CST-APCA outperforms all compared approaches, and significantly reduces the error rate to 3.02%.

7.4.8 Computational Complexity Analysis

In this section, we compare the time consumption of different approaches on the high-dim LBP feature of the LFW dataset, including time consumption of both training and testing. PmSVM is implemented in C++, and others are implemented in Matlab. The PC has Intel(R) Core(TM) i7-3930K CPU@ 3.2GHz and 16G RAM. The results are shown in Table 7.7.

Table 7.7: Comparison of time consumption of different approaches on the high-dim LBP features of the LFW database.

Algorithms	Time (s)
NNC-ED	19
NNC-Chi2D	21
PCA-mMDC	1295
SR-PCA-mMDC [175]	1618
PmSVM-Chi2 [174]	14217
PmSVM-HI [174]	14695
Proposed CST-APCA	5428

As expected, NNC requires least time as no training required. SR-PCA-mMDC requires more time than PCA-mMDC due to square-rooting. The proposed approach requires more time than other subspace approaches, mainly due to the memory limitations so that we have to save data to harddisk and reload them later. Even so, the proposed approach is still much faster than PmSVMs. Note that the evaluation is 10-fold cross-validation on high-dim LBP features (127,440 dimensions) of the LFW dataset (13,233 images).

7.4.9 Summary and Discussion

From all 11 aforementioned image-classification results, it is very interesting and informative to see the following facts:

1. Without dimensionality reduction, Chi-squared distance consistently outperforms Euclidean distance.
2. The dimension-reduction approaches such as PCA-mMDC, SR-PCA-mMDC and the proposed CST-APCA consistently outperform NNC-ED.
3. Without the proposed transformation, dimensionality-reduction approaches do not always outperform NNC-Chi2D, e.g. PCA-mMDC performs poorer than NNC-Chi2D on the AR database.
4. SR-PCA-mMDC outperforms PCA-mMDC consistently.
5. The proposed approach consistently leads to the best performance in 10 out of 11 cases. Some performance gains are significant.
6. Besides LBP features, the proposed approach also outperforms other approaches on HOG features of the LFW dataset.

The above facts clearly lead to the following analysis:

1. Infrequent patterns play an important role in LBP-histogram matching. Chi-squared distance better handles them than Euclidean distance.
2. Subspace approaches remove the unreliable dimensions of LBP features, and hence improve the performance.
3. LBP features deviate far from Gaussian distribution, which limits the performance gain brought by subspace approaches.

4. Square-rooting the features improves the fitness to Gaussian, and hence improves the performance.
5. The proposed Chi-squared transformation produces a feature that fits better to Gaussian distribution, and hence enhances the performance gain of subspace approaches.
6. The proposed transformation improves the performance not only on LBP features, but also on other histogram-like features such as HOG.

7.5 Summary

Subspace approaches have been applied to the LBP histogram in order to remove unreliable dimensions or to derive a compact feature representation. We show that LBP-histogram feature deviates far from Gaussian distribution, which largely limits the performance gain brought by subspace approaches. This work proposes a truncated Gaussian model for the LBP feature, and a Chi-squared transformation to reduce the truncation effect. Both analysis and experiments show that the transformed feature fits better to Gaussian distribution. The proposed transformation naturally leads to an asymmetric two-class classification problem. We propose a CST-APCA approach to better remove the unreliable dimensions in the CST feature space. The proposed approach is tested on face recognition, protein cellular classification and dynamic texture classification. All experiments have demonstrated consistent performance gains brought by the proposed Chi-squared transformation, some of which are significant.

Chapter 8

Conclusions and Future Work

In this thesis, we study subspace approaches on local binary patterns, and aim to exploit the advantages of both approaches for visual recognition. This chapter is organized as follows. In Section 8.1, our contributions of subspace approaches on traditional holistic features are presented. The contributions of improving the robustness to image noise and reducing the dimensionality of LBP features are given in Section 8.2. The contributions of removing the unreliable dimensions and handling the non-Gaussian distribution of LBP features are discussed in Section 8.3. Finally, Some potential research directions are discussed in Section 8.4.

8.1 Contributions on Subspace Approaches on Holistic Features

For subspace approaches on holistic features, rigorous geometric normalization is often required to achieve a good recognition performance, e.g. an accurate face/eye detector for face alignment is crucial for face recognition. Traditional subspace approaches are not suitable for detection tasks due to a huge amount of scanning windows involved and large-matrix multiplication required. We propose a cascade APCDA face/eye detector that locates the eyes at a speed comparable to Adaboost face/eye detector. The proposed face/eye detector can discover

more discriminative features, and hence locate the eye at a much higher precision than Adaboost face/eye detector. We then develop a face verification system on mobile device using subspace approaches. We test various subspace approaches and ERE approach demonstrates a superior performance to others. The proposed approach class-specific threshold without using validation data further enhances the performance.

8.2 Contributions on Improvements of LBP Features

Subspace approaches on holistic features often suffers from two constraints: 1) Rigorous image alignment required. However, for many tasks such as scene recognition, such a rigorous image alignment is not feasible. 2) Large-matrix multiplication involved. Holistic features may have a high feature dimensionality by vectorizing image raw pixels. Thus, the images are often down-sized, which may result in a significant information loss.

Recently, many features such as SIFT, HOG and LBP are proposed for visual recognition tasks. LBP feature is one of most popular features. In this thesis, we improve the LBP features from two aspects: 1) As LBP feature is sensitive to image noise, we propose a noise-resistant local binary pattern to address this issue. 2) As LBP feature is often of high dimensionality, we propose a novel framework of reducing the dimensionality of LBP features by optimizing the LBP structure directly.

8.2.1 Noise-Resistant Local Binary Pattern

LBP feature is sensitive to image noise as a small image variation may alter the LBP code significantly. Local ternary pattern and its variants partially solve this problem, but they do not attempt to recover the corrupted patterns. We thus propose a noise-resistant local binary pattern with an embedded error-correction mechanism. The small image pixel differences are easily distorted by noise. Thus, we encode them separately as “uncertain code”, and later

determine their values using other “curtain” bits. It is well-known that image micro-structures are mainly captured by uniform patterns. We thus correct the corrupted LBP codes by assigning the values of “uncertain” bits to form only uniform codes. We further find that some image micro-structures such as line patterns are grouped as non-uniform patterns and ignored. Those line patterns may occur less frequent than uniform patterns, but they represent an important group of image primitives. Thus, we propose to represent the line patterns by an extended set of uniform patterns, and propose an extended NRLBP feature to recover these line patterns. The proposed approaches are validated extensively on various recognition tasks.

8.2.2 Dimensionality Reduction by Optimizing LBP Structure

LBP features are often extracted from multiple image patches at multiple scales, which will result in a high-dimensional feature vector. Recently, high-dimensional LBP features have demonstrated a superior performance on face recognition. However, it needs to be reduced to a reasonable size before further processing. Many feature selection/extraction techniques have been proposed to address this issue. However, they need to enumerate all the histogram bins first, which may not be feasible for a large LBP structure. Thus, the large LBP structures are often broken into small handcrafted ones. However, the optimal LBP structures should be task-dependent. Furthermore, the handcrafted structures may not be optimal as they are often selected heuristically. We thus propose a new framework to derive the optimal LBP structure, instead of direct bin selection/extraction or utilizing handcrafted structures. We cast the LBP structure optimization as a point selection problem, and propose a Maximal-Conditional-Mutual-Information scheme to handle strong correlation of binarized-pixel-difference features. The propose approach demonstrates a superior performance on various recognition tasks.

8.3 Contributions on Subspace Approaches on LBP Features

LBP histogram may not be reliably estimated due to insufficient pixels and its unbalance nature. Thus, many histogram bins are noisy and unreliable. We propose two learning-based LBP features to handle this problem. A more severe problem of LBP-histogram feature is its non-Gaussian distribution. To tackle this, we propose a Chi-squared transformation so that the transformed feature fits better to Gaussian.

8.3.1 Learn-based Local Binary Patterns

To suppress the noise in LBP histogram, all the noisy non-uniform patterns are grouped into one bin in uniform LBP. We argue that useful patterns are task-dependent and non-uniform patterns may also be useful for classification in some scenarios. Thus, we propose two learning-based LBP features: PI-LBP and PD-LBP by applying the principal histogram analysis on LBP histogram to remove the unreliable information. We further find that for dynamic texture classification, spatial information is more important than temporal information. Thus, we propose a “super histogram” to better capture the spatial information of dynamic textures. The proposed learn-based LBP features are validated on a widely range of applications. Typically, the “super histogram” on the proposed PI-LBP and PD-LBP outperforms the state-of-the-art approaches on dynamic texture recognition significantly.

8.3.2 A Chi-squared Transformation on LBP Features

Subspace approaches are often applied on the LBP histogram, and they are optimal under Gaussian assumption. However, we find that due to non-negative and simplex constraints, LBP feature does not follow Gaussian distribution, which limits the performance of subspace approaches. Thus, we propose a Chi-squared transformation to transfer the LBP-histogram features into a new feature space. Both our analysis and experiments show that the distribution

of the transformed feature fits closer to Gaussian. Our formulation naturally leads to a two-class classification problem. This problem is high asymmetric. Thus, we propose a CST-APCA approach for LBP-histogram features. Extensive experimental results on various visual recognition tasks demonstrate the effectiveness of the proposed transformation.

8.4 Future Work

In future, we plan to further improve the robustness of local binary pattern and apply the proposed techniques on ordinal features.

8.4.1 An Improved Noise-Resistant Local Binary Pattern

For noise-resistant LBP, we make use of the prior knowledge that uniform patterns represent the image micro-structures and they more likely occur in a natural image than non-uniform patterns. We treat every uniform code equally, i.e. when multiple uniform codes can be formed, we assign an equal weight for every code. In fact, we could improve this by embedding more prior knowledge into the encoding scheme. One possible solution is that we first learn a prior distribution of local binary patterns on images without noise (or with very little noise), then we utilize both the prior distribution of the LBP codes and the likelihood of a pixel difference to be encoded as “0” or “1” to determine the posterior probability of a code for a set of given pixel differences. In such a way, we incorporate our prior knowledge into the encoding process. This improved NRLBP is expected more resistant to image noise.

8.4.2 Learning Discriminative Patch-Wise LBP Structures

In Chapter 5, we learn the LBP structure in a data-driven way, but we still utilize some pre-defined framework to extract LBP features, i.e. we utilize patches extracted in a grid or hierarchical structure. We could further relax the constraint by learning discriminative patch-wise

LBP structures in a data-driven way. Instead of utilizing the handcrafted patch locations and handcrafted LBP structures, we can derive the most discriminative LBP structures at the optimal patch locations. We plan to even include the possible scale variations in the optimization process so that we could determine the optimal scale at the same time.

8.4.3 A Chi-squared transformed Data-drive NRLBP

In this thesis, we improve the performance of LBP features from different aspects. i.e. NRLBP that is more resistant to image noise, the data-driven LBP that can find the most suitable LBP structure for the specific application and Chi-squared transformation that can improve the performance gain of subspace approaches on LBP features. However, we have not studied the performance gain when these techniques are combined. It will be interesting to fuse these techniques. It is expected that the performance gain will be significantly larger than only using one of the approaches. We may also investigate how to fuse these techniques in the best way.

8.4.4 Learning Optimal Structure for Local Ordinal Pattern

Ordinal features are also widely used in visual recognition. However, the dimensionality of ordinal features is even higher than that of LBP features when utilizing the same number of neighbors, i.e. the number of its histogram bins is $n!$, where n is the number of neighbors. Therefore, it is even more critical to learn an optimal ordinal structures. We can investigate several possibilities to solve this problem. One possible solution is to learn the optimal structures for ordinal features in a data-driven way, with necessary adaptations.

8.4.5 Optimizing Multi-channel LBP Structure

LBP features can be extracted from multiple channels. For example, LBP features extracted from multiple color channels have demonstrated a good performance on scene classification. In order to exploit the coherence information from different color channels, LBP features should

be extracted cross different color channels. However, the structure optimization proposed in Chapter 5, which selects a subset of points to compose one optimal LBP structure, may not be suitable to solve this problem. The optimal LBP structure cannot be too small, otherwise it is not discriminative enough. It is can neither be too large, as it is very difficult to handle a large LBP structure. Instead, we may select a set of sub-optimal LBP structures, which as a whole optimally characterize the multi-channel LBP structure.

8.4.6 Improve the Fitness to Gaussian for Histogram-like Features

In Chapter 7, we propose a Chi-squared transformation to improve the fitness to Gaussian. The experimental results show that by improving the fitness, Chi-squared transformation successfully improves the performance of subspace approaches. Although we have analyzed the proposed Chi-squared transformation both theoretically and experimentally, the problem is not solved from the point of view of improving the fitness. In another word, in future we plan to study how to transform a histogram-like feature with a modelled distribution to Gaussian distribution. Chi-squared transformation is one of possible solutions, but there are maybe other solutions that could perform better.

8.4.7 Subspace Approach on Large Dataset

Due to the computational complexity and storage constraints, the dimensionality of subspace approach is often limited to 10,000. However, with the increasing trends of large data, the data set now has much more samples than before. The number of samples may easily reach to 1 million. In literature, as well as in this thesis, it has been shown that high-dim LBP features can achieve a much better performance than baseline LBP. Thus, it imposes a huge challenge to work with a large dataset with over 100,000 training samples of over 100,000 feature dimensions. Under this setting, many existing subspace approaches are not feasible as it is even difficult to build the covariance matrix. Our future plan includes how to apply subspace ap-

proaches on high-dim LBP with such a huge amount of training samples.

Author's Publications

Journal Papers

1. **Jianfeng Ren**, Xudong Jiang and Junsong Yuan, "A Complete and Fully Automated Face Verification System on Mobile Devices", *Pattern Recognition*, vol. 46, no. 1, pp. 45-56, January 2013.
2. **Jianfeng Ren**, Xudong Jiang and Junsong Yuan, "Noise-Resistant Local Binary Pattern with an Embedded Error-Correction Mechanism", *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 4049-4060, October, 2013.
3. **Jianfeng Ren**, Xudong Jiang, Junsong Yuan and Gang Wang, "Optimizing LBP Structure For Visual Recognition Using Binary Quadratic Programming", *IEEE Signal Processing Letters*, vol. 21, no. 11, pp. 1346-1350, 2014.
4. **Jianfeng Ren**, Xudong Jiang and Junsong Yuan, "Learning LBP Structure by Maximizing the Conditional Mutual Information", *Pattern Recognition*, DOI: 10.1016/j.patcog.2015.02.001, online published, 2015.
5. **Jianfeng Ren**, Xudong Jiang and Junsong Yuan, "A Chi-Squared-Transformed Subspace of LBP Histogram for Visual Recognition", *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1893-1904, June, 2015.

Conference Papers

1. **Jianfeng Ren** and Xudong Jiang, "Fast Eye Localization Based on Pixel Differences", *IEEE International Conference on Image Processing (ICIP)*, pp. 2733-2736, Cairo, Egypt, November, 2009.
2. **Jianfeng Ren** and Xudong Jiang, "Eye Detection Based on Rank Order Filter", *IEEE 7th International Conference on Information, Communications and Signal Processing*, Macau, December, 2009.

3. **Jianfeng Ren**, Xudong Jiang and Junsong Yuan, "A Fast and Accurate Cascade Subspace Face/Eye Detector on Mobile Devices", *International Conference on Computer Vision (ICCV2011), Workshop on Mobile Vision*, pp. 84-91, Barcelona, Spain, November 2011.
4. **Jianfeng Ren**, Xudong Jiang and Junsong Yuan, "Dynamic Texture Recognition Using Enhanced LBP Features", *IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2400-2404, Vancouver, Canada, May, 2013.
5. **Jianfeng Ren**, Xudong Jiang and Junsong Yuan, "Relaxed Local Ternary Pattern for Face Recognition", *IEEE Int. Conf. Image Processing (ICIP)*, pp. 3680-3684, Melbourne, Australia, September, 2013.
6. **Jianfeng Ren**, Xudong Jiang and Junsong Yuan, "Learning Binarized Pixel-difference Pattern for Scene Recognition", *IEEE Int. Conf. Image Processing (ICIP)*, pp. 2494-2498, Melbourne, Australia, September, 2013.
7. **Jianfeng Ren**, Xudong Jiang and Junsong Yuan, "Quantized Fuzzy LBP for Face Recognition", *IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2015, accepted.

Bibliography

- [1] X. Tan and B. Triggs, “Enhanced local texture feature sets for face recognition under difficult lighting conditions,” *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1635–1650, 2010.
- [2] S. ul Hussain and B. Triggs, “Visual recognition using local quantized patterns,” in *European Conference on Computer Vision*, pp. 716–729, Springer, 2012.
- [3] J. Wu and J. Rehg, “CENTRIST: A visual descriptor for scene categorization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1489–1501, 2011.
- [4] F. Fleuret, “Fast binary feature selection with conditional mutual information,” *Journal of Machine Learning Research*, vol. 5, pp. 1531–1555, 2004.
- [5] Y. Xu, Y. Quan, H. Ling, and H. Ji, “Dynamic texture classification using dynamic fractal analysis,” in *International Conference on Computer Vision*, pp. 1219–1226, nov. 2011.
- [6] M. Turk and A. Pentland, “Face recognition using eigenfaces,” in *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–591, 1991.
- [7] P. Belhumeur, J. Hespanha, and D. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.
- [8] W. Liu, Y. Wang, S. Z. Li, and T. Tan, “Null space approach of fisher discriminant analysis for face recognition,” in *Biometric Authentication*, pp. 32–44, Springer, 2004.
- [9] X. Wang and X. Tang, “Dual-space linear discriminant analysis for face recognition,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. II–564, IEEE, 2004.
- [10] B. Moghaddam, “Principal manifolds and probabilistic subspaces for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 780–788, 2002.

- [11] X. Jiang, B. Mandal, and A. Kot, "Enhanced maximum likelihood face recognition," *Electronics Letters*, vol. 42, no. 19, pp. 1089–1090, 2006.
- [12] X. Jiang, B. Mandal, and A. Kot, "Eigenfeature regularization and extraction in face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 383–394, 2008.
- [13] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.
- [14] D. G. Lowe, "Object recognition from local scale-invariant features," in *IEEE international conference on Computer vision*, vol. 2, pp. 1150–1157, Ieee, 1999.
- [15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, IEEE, 2005.
- [16] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
- [17] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.
- [18] S. U. Hussain, T. Napoléon, F. Jurie, *et al.*, "Face recognition using local quantized patterns," in *British Machine Vision Conference*, 2012.
- [19] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 915–928, 2007.
- [20] R. Gupta, H. Patil, and A. Mittal, "Robust order-based methods for feature description," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 334–341, 2010.
- [21] S. Yan, H. Wang, X. Tang, and T. Huang, "Exploring feature descriptors for face recognition," in *Proceeding of IEEE International. Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 629–632, 2007.
- [22] Y. Fang and Z. Wang, "Improving LBP features for gender classification," in *Proceeding of International. Conference on Wavelet Analysis and Pattern Recognition*, pp. 373–377, 2008.

- [23] B. Ghanem and N. Ahuja, "Maximum margin distance learning for dynamic texture recognition," in *European Conference on Computer Vision*, pp. 223–236, 2010.
- [24] X. Jiang, "Asymmetric principal component and discriminant analyses for pattern classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 931–937, 2009.
- [25] X. Jiang, "Linear subspace learning-based dimensionality reduction," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 16–26, 2011.
- [26] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, 2010.
- [27] J. Ren, X. Jiang, and J. Yuan, "Noise-resistant local binary pattern with an embedded error-correction mechanism," *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 4049–4060, 2013.
- [28] B. Moghaddam, T. Jebara, and A. Pentland, "Bayesian face recognition," *Pattern Recognition*, vol. 33, no. 11, pp. 1771–1782, 2000.
- [29] S. Liao, M. Law, and A. Chung, "Dominant local binary patterns for texture classification," *IEEE Transactions on Image Processing*, vol. 18, no. 5, pp. 1107–1118, 2009.
- [30] Z. Li, G. Liu, Y. Yang, and J. You, "Scale-and rotation-invariant local binary pattern using scale-adaptive texton and subuniform-based circular shift," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2130–2140, 2012.
- [31] G. Zhao, T. Ahonen, J. Matas, and M. Pietikainen, "Rotation-invariant image and video description with local binary pattern features," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1465–1477, 2012.
- [32] J. Ren, X. Jiang, and J. Yuan, "Dynamic texture recognition using enhanced lbp features," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2400–2404, 2013.
- [33] J. Ren, X. Jiang, and J. Yuan, "Relaxed local ternary pattern for face recognition," in *IEEE International Conference on Image Processing (ICIP)*, pp. 3680–3684, 2013.
- [34] J. Ren, X. Jiang, and J. Yuan, "Learning binarized pixel-difference pattern for scene recognition," in *IEEE International Conference on Image Processing (ICIP)*, pp. 2494–2498, 2013.

- [35] Y. Xiao, J. Wu, and J. Yuan, "mCENTRIST: A multi-channel feature generation mechanism for scene categorization," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 823–836, 2014.
- [36] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognition*, vol. 42, no. 3, pp. 425–436, 2009.
- [37] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikainen, and S. Li, "Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1301–1306, IEEE, 2010.
- [38] A. Satpathy, X. Jiang, and H. Eng, "Lbp based edge-texture features for object recognition," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 1953–1964, 2014.
- [39] H. Zhou, R. Wang, and C. Wang, "A novel extended local-binary-pattern operator for texture analysis," *Information Sciences*, vol. 178, no. 22, pp. 4314–4325, 2008.
- [40] A. Fathi and A. Naghsh-Nilchi, "Noise tolerant local binary pattern operator for efficient texture analysis," *Pattern Recognition Letters*, 2012.
- [41] L. Nanni, S. Brahmam, and A. Lumini, "A simple method for improving local binary patterns by considering non-uniform patterns," *Pattern Recognition*, vol. 45, pp. 3844–3852, Oct. 2012.
- [42] T. Ahonen and M. Pietikäinen, "Soft histograms for local binary patterns," in *Proceedings of the Finnish signal processing symposium (FINSIG 2007)*, pp. 1–4, 2007.
- [43] D. Iakovidis, E. Keramidas, and D. Maroulis, "Fuzzy local binary patterns for ultrasound texture characterization," *Image Analysis and Recognition*, pp. 750–759, 2008.
- [44] N. Tan, L. Huang, and C. Liu, "A new probabilistic local binary pattern for face verification," in *Proceeding of IEEE International Conference on Image Processing*, pp. 1237–1240, 2009.
- [45] E. Keramidas, D. Iakovidis, and D. Maroulis, "Fuzzy binary patterns for uncertainty-aware texture representation," *Electronic Letters on Computer Vision and Image Analysis*, vol. 10, no. 1, pp. 63–78, 2011.
- [46] L. Nanni, A. Lumini, and S. Brahmam, "Local binary patterns variants as texture descriptors for medical image analysis," *Artificial intelligence in medicine*, vol. 49, no. 2, pp. 117–125, 2010.

- [47] M. Akhloufi and A. Bendada, "Locally adaptive texture features for multispectral face recognition," in *IEEE International Conference on Systems Man and Cybernetics (SMC)*, pp. 3308–3314, IEEE, 2010.
- [48] W. Liao and T. Young, "Texture classification using uniform extended local ternary patterns," in *2010 IEEE International Symposium on Multimedia (ISM)*, pp. 191–195, IEEE, 2010.
- [49] D. He and N. Cercone, "Local triplet pattern for content-based image retrieval," *Image Analysis and Recognition*, pp. 229–238, 2009.
- [50] A. Hafiane, G. Seetharaman, and B. Zavidovique, "Median binary pattern for textures classification," in *Image Analysis and Recognition*, pp. 387–398, Springer, 2007.
- [51] B. Froba and A. Ernst, "Face detection with the modified census transform," in *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pp. 91–96, IEEE, 2004.
- [52] J. Trefny and J. Matas, "Extended set of local binary patterns for rapid object detection," in *Proceedings of the Computer Vision Winter Workshop*, vol. 2010, 2010.
- [53] X. Fu and W. Wei, "Centralized binary patterns embedded with image euclidean distance for facial expression recognition," in *International Conference on Natural Computation*, vol. 4, pp. 115–119, IEEE, 2008.
- [54] C. He, T. Ahonen, and M. Pietikainen, "A bayesian local binary pattern texture descriptor," in *International Conference on Pattern Recognition*, pp. 1–4, IEEE, 2008.
- [55] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition," in *IEEE International Conference on Computer Vision*, vol. 1, pp. 786–791, IEEE, 2005.
- [56] C.-H. Yao and S.-Y. Chen, "Retrieval of translated, rotated and scaled color textures," *Pattern Recognition*, vol. 36, no. 4, pp. 913–929, 2003.
- [57] X. Huang, S. Z. Li, and Y. Wang, "Shape localization based on statistical method using extended local binary pattern," in *IEEE First Symposium on Multi-Agent Security and Survivability*, pp. 184–187, IEEE, 2004.
- [58] S. Zhao, Y. Gao, and B. Zhang, "Sobel-LBP," in *IEEE International Conference on Image Processing*, pp. 2144–2147, IEEE, 2008.
- [59] X. Li, W. Hu, Z. Zhang, and H. Wang, "Heat kernel based local binary pattern for face representation," *IEEE Signal Processing Letters*, vol. 17, no. 3, pp. 308–311, 2010.

- [60] B. Li and M. Q.-H. Meng, "Texture analysis for ulcer detection in capsule endoscopy images," *Image and Vision computing*, vol. 27, no. 9, pp. 1336–1342, 2009.
- [61] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li, "Learning multi-scale block local binary patterns for face recognition," in *Advances in Biometrics*, pp. 828–837, Springer, 2007.
- [62] Y. He, N. Sang, and C. Gao, "Pyramid-based multi-structure local binary pattern for texture classification," in *Asian Conference on Computer Vision*, pp. 133–144, Springer, 2011.
- [63] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3025–3032, IEEE, 2013.
- [64] O. Lahdenoja, M. Laiho, and A. Paasio, "Reducing the feature vector length in local binary pattern based face recognition," in *IEEE International Conference on Image Processing*, vol. 2, pp. II–914, IEEE, 2005.
- [65] H. Yang and Y. Wang, "A LBP-based face recognition method with hamming distance constraint," in *International Conference on Image and Graphics*, pp. 645–649, IEEE, 2007.
- [66] M. Topi, O. Timo, P. Matti, and S. Maricor, "Robust texture classification by subsets of local binary patterns," in *International Conference on Pattern Recognition*, vol. 3, pp. 935–938, IEEE, 2000.
- [67] R. S. Smith and T. Windeatt, "Facial expression detection using filtered local binary pattern features with ecoc classifiers and platt scaling," *Journal of Machine Learning Research*, pp. 111–118, 2010.
- [68] Y. Guo, G. Zhao, M. Pietikäinen, and Z. Xu, "Descriptor learning based on fisher separation criterion for texture classification," in *Asian Conference on Computer Vision*, pp. 185–198, Springer, 2011.
- [69] S. Z. Li, C. Zhao, M. Ao, and Z. Lei, "Learning to fuse 3d+ 2d based face recognition at both feature and decision levels," in *Analysis and Modelling of Faces and Gestures*, pp. 44–54, Springer, 2005.
- [70] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Li, "Face detection based on multi-block LBP representation," *Advances in Biometrics*, pp. 11–18, 2007.
- [71] C. Shan and T. Gritti, "Learning discriminative lbp-histogram bins for facial expression recognition," in *British Machine Vision Conference*, pp. 1–10, 2008.

- [72] C. Heng, S. Yokomitsu, Y. Matsumoto, and H. Tamura, "Shrink boost for selecting multi-lbp histogram features in object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3250–3257, 2012.
- [73] S. Liao and A. C. Chung, "Face recognition by using elongated local binary patterns with average maximum distance gradient magnitude," in *Asian Conference on Computer Vision*, pp. 672–679, Springer, 2007.
- [74] A. Petpon and S. Srisuk, "Face recognition with local line binary pattern," in *International Conference on Image and Graphics*, pp. 533–539, IEEE, 2009.
- [75] M. Turtinen and M. Pietikäinen, "Contextual analysis of textured scene images.," in *British Machine Vision Conference*, pp. 849–858, Citeseer, 2006.
- [76] L. Wolf, T. Hassner, Y. Taigman, *et al.*, "Descriptor based methods in the wild," in *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, 2008.
- [77] Y. Mu, S. Yan, Y. Liu, T. Huang, and B. Zhou, "Discriminative local binary patterns for human detection in personal album," in *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.
- [78] D. Maturana, D. Mery, and A. Soto, "Learning discriminative local binary patterns for face recognition," in *IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011)*, pp. 470–475, IEEE, 2011.
- [79] Z. Lei, M. Pietikainen, and S. Li, "Learning discriminant face descriptor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, pp. 289–302, Feb 2014.
- [80] N. Sun, W. Zheng, C. Sun, C. Zou, and L. Zhao, "Gender classification based on boosting local binary pattern," in *Advances in Neural Networks*, pp. 194–201, Springer, 2006.
- [81] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [82] J. Yuan, M. Yang, and Y. Wu, "Mining discriminative co-occurrence patterns for visual recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2777–2784, 2011.
- [83] Z. Cao, Q. Yin, X. Tang, and J. Sun, "Face recognition with learning-based descriptor," in *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2707–2714, 2010.

- [84] W. Wang, F. Chang, J. Zhao, and Z. Chen, "Automatic facial expression recognition using local binary pattern," in *World Congress on Intelligent Control and Automation (WCICA)*, pp. 6375–6378, IEEE, 2010.
- [85] C. Zeng and H. Ma, "Robust head-shoulder detection by pca-based multilevel HOG-LBP detector for people counting," in *International Conference on Pattern Recognition (ICPR)*, pp. 2069–2072, IEEE, 2010.
- [86] C.-H. Chan, J. Kittler, and K. Messer, "Multi-scale local binary pattern histograms for face recognition," in *Advances in Biometrics* (S.-W. Lee and S. Li, eds.), vol. 4642 of *Lecture Notes in Computer Science*, pp. 809–818, Springer Berlin Heidelberg, 2007.
- [87] K. H. An and M. J. Chung, "Cognitive face analysis system for future interactive tv," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 2271–2279, 2009.
- [88] S. Shan, W. Zhang, Y. Su, X. Chen, and W. Gao, "Ensemble of piecewise fda based on spatial histograms of local (gabor) binary patterns for face recognition," in *International Conference on Pattern Recognition*, vol. 4, pp. 606–609, IEEE, 2006.
- [89] D. Zhao, Z. Lin, and X. Tang, "Contextual distance for data perception," in *International Conference on Computer Vision*, pp. 1–8, IEEE, 2007.
- [90] D. Zhao, Z. Lin, and X. Tang, "Laplacian pca and its applications," in *International Conference on Computer Vision*, pp. 1–8, IEEE, 2007.
- [91] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.
- [92] S. U. Hussain, W. Triggs, *et al.*, "Feature sets and dimensionality reduction for visual object detection," in *British Machine Vision Conference*, pp. 112.1–112.10, 2010.
- [93] C. Shan, S. Gong, and P. W. McOwan, "Appearance manifold of facial expression," in *Computer Vision in Human-Computer Interaction*, pp. 221–230, Springer, 2005.
- [94] P. Phillips, H. Moon, S. Rizvi, and P. Rauss, "The feret evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.
- [95] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 210–227, 2009.
- [96] D.-Q. Dai and P. C. Yuen, "Regularized discriminant analysis and its application to face recognition," *Pattern Recognition*, vol. 36, no. 3, pp. 845–847, 2003.

- [97] L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu, "A new lda-based face recognition system which can solve the small sample size problem," *Pattern recognition*, vol. 33, no. 10, pp. 1713–1726, 2000.
- [98] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.
- [99] A. M. Martinez, "The ar face database," *CVC Technical Report*, vol. 24, 1998.
- [100] W. Gao, B. Cao, S. Shan, X. Chen, D. Zhou, X. Zhang, and D. Zhao, "The cas-peal large-scale chinese face database and baseline evaluations," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 38, no. 1, pp. 149–161, 2008.
- [101] S.-U. Jung, Y.-S. Chung, J.-H. Yoo, and K.-Y. Moon, "Real-time face verification for mobile platforms," in *Advances in visual computing*, pp. 823–832, Springer, 2008.
- [102] P. Abeni, M. Baltatu, and R. DAlessandro, "A face recognition system for mobile phones," in *ISSE 2006 Securing Electronic Business Processes*, pp. 211–217, Springer, 2006.
- [103] Q. Tao and R. Veldhuis, "Biometric authentication system on mobile personal devices," *Instrumentation and Measurement, IEEE Transactions on*, vol. 59, no. 4, pp. 763–773, 2010.
- [104] A. Hadid, J. Heikkilä, O. Silvén, and M. Pietikainen, "Face and eye detection for person authentication in mobile phones," in *ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 101–108, IEEE, 2007.
- [105] C. K. Ng, M. Savvides, and P. K. Khosla, "Real-time face verification system on a cell-phone using advanced correlation filters," in *IEEE Workshop on Automatic Identification Advanced Technologies*, pp. 57–62, IEEE, 2005.
- [106] C. Geng and X. Jiang, "Face recognition based on the multi-scale local image structures," *Pattern Recognition*, vol. 44, no. 10, pp. 2565–2575, 2011.
- [107] S. Jassim, H. Sellaheewa, and J.-H. Ehlers, "Wavelet-based face verification for mobile personal devices," *Biometrics on the Internet*, pp. 81–84, 2005.
- [108] K. Choi, K.-A. Toh, and H. Byun, "Realtime training on mobile devices for face recognition applications," *Pattern recognition*, vol. 44, no. 2, pp. 386–400, 2011.

- [109] K. Venkataramani, S. Qidwai, and B. Vijayakumar, "Face authentication from cell phone camera images with illumination and temporal variations," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 35, no. 3, pp. 411–418, 2005.
- [110] J. Yang, X. Chen, W. Kunz, and H. Kundra, "Face as an index: Knowing who is who using a pda," *International journal of imaging systems and technology*, vol. 13, no. 1, pp. 33–41, 2003.
- [111] H. Yu and J. Yang, "A direct lda algorithm for high-dimensional data with application to face recognition," *Pattern recognition*, vol. 34, no. 10, pp. 2067–2070, 2001.
- [112] C. Schneider, N. Esau, L. Kleinjohann, and B. Kleinjohann, "Feature based face localization and recognition on mobile devices," in *International Conference on Control, Automation, Robotics and Vision*, pp. 1–6, IEEE, 2006.
- [113] P.-F. Hsieh, D.-S. Wang, and C.-W. Hsu, "A linear feature extraction for multiclass classification problems based on class mean and covariance discriminant information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 223–235, 2006.
- [114] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bulletin of the Calcutta Mathematical Society*, vol. 35, pp. 99–109, 1943.
- [115] K. Fukunaga, *Introduction to statistical pattern recognition*. Academic press, 1990.
- [116] S. Yan, D. Xu, and X. Tang, "Face verification with balanced thresholds," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 262–268, 2007.
- [117] G. Goudelis, S. Zafeiriou, A. Tefas, and I. Pitas, "Class-specific kernel-discriminant analysis for face verification," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 570–587, 2007.
- [118] G. Bradski, "The opencv library," *Doctor Dobbs Journal*, vol. 25, no. 11, pp. 120–126, 2000.
- [119] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *International Conference on Image Processing*, vol. 1, pp. I–900, IEEE, 2002.
- [120] M.-T. Pham and T.-J. Cham, "Fast training and selection of haar features using statistics in boosting-based face detection," in *International Conference on Computer Vision*, pp. 1–7, IEEE, 2007.

- [121] M.-T. Pham, V. Hoang, and T.-J. Cham, "Detection with multi-exit asymmetric boosting," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.
- [122] S. Hameed, "Tree-based 20×20 frontal eye detector with better handling of eyeglasses," <http://umich.edu/~shameem>.
- [123] H. Yu, "Tree-based 20×20 left eye detector," <http://yushiqi.cn/research/eyedetection>.
- [124] M. Castrillón, O. Déniz, C. Guerra, and M. Hernández, "Encara2: Real-time detection of multiple faces at different resolutions in video streams," *Journal of Visual Communication and Image Representation*, vol. 18, no. 2, pp. 130–140, 2007.
- [125] G. Li, X. Cai, X. Li, and Y. Liu, "An efficient face normalization algorithm based on eyes detection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3843–3848, 2006.
- [126] D.-T. Lin and C.-M. Yang, "Real-time eye detection using face-circle fitting and dark-pixel filtering," in *IEEE International Conference on Multimedia and Expo (ICME)*, vol. 2, pp. 1167–1170, 2004.
- [127] J. Ren and X. Jiang, "Eye detection based on rank order filter," in *International Conference on Information, Communications and Signal Processing (ICICS)*, pp. 1–4, 2009.
- [128] J. Ren and X. Jiang, "Fast eye localization based on pixel differences," in *IEEE International Conference on Image Processing (ICIP)*, pp. 2733–2736, 2009.
- [129] <http://users.ece.gatech.edu/~justin/llmagic/>.
- [130] <http://www.cis.eee.ntu.edu.sg/Projects/Pages/FaceVerificationSystemonMobileDevices.aspx>.
- [131] M. Inen, M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen, *Computer Vision Using Local Binary Patterns*, vol. 40. Springer Verlag, 2011.
- [132] A. Martinez and R. Benavente, "The AR face database," *CVC Tech. Report #24*, 1998.
- [133] T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, and S. Huovinen, "Outex-new framework for empirical evaluation of texture analysis algorithms," in *Proceedings of International Conference on Pattern Recognition*, vol. 1, pp. 701–706, 2002.
- [134] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

- [135] K. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.
- [136] J. Ren, X. Jiang, and J. Yuan, "A complete and fully automated face verification system on mobile devices," *Pattern Recognition*, vol. 46, no. 1, pp. 45 – 56, 2013.
- [137] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.
- [138] M. V. Boland and R. F. Murphy, "A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of hela cells," *Bioinformatics*, vol. 17, no. 12, pp. 1213–1223, 2001.
- [139] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [140] Y. Guo, G. Zhao, and M. Pietikinen, "Discriminative features for texture description," *Pattern Recognition*, vol. 45, no. 10, pp. 3834 – 3843, 2012.
- [141] Z. Lei, M. Pietikainen, and S. Li, "Learning discriminant face descriptor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, pp. 289–302, Feb 2014.
- [142] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.
- [143] J. Ren, X. Jiang, J. Yuan, and G. Wang, "Optimizing lbp structure for visual recognition using binary quadratic programming," *IEEE Signal Processing letters*, vol. 21, no. 11, pp. 1346 – 1350, 2014.
- [144] A. Jain, R. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.
- [145] C. Shan and T. Gritti, "Learning discriminative LBP-histogram bins for facial expression recognition," in *In Proc. British Machine Vision Conference*, 2008.
- [146] C. Shan, "Learning local binary patterns for gender classification on real-world face images," *Pattern Recognition Letters*, vol. 33, no. 4, pp. 431–437, 2012.
- [147] G. Brown, "An information theoretic perspective on multiple classifier systems," in *Multiple Classifier Systems*, pp. 344–353, Springer, 2009.

- [148] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.
- [149] Y. Yang and S. Newsam, "Spatial pyramid co-occurrence for image classification," in *International Conference on Computer Vision*, pp. 1465–1472, 2011.
- [150] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto, "Dynamic textures," *International Journal of Computer Vision*, vol. 51, no. 2, pp. 91–109, 2003.
- [151] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2169–2178, 2006.
- [152] B. Caputo, E. Hayman, and P. Mallikarjuna, "Class-specific material categorisation," in *IEEE International Conference on Computer Vision*, pp. 1597–1604, 2005.
- [153] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, and W. Gao, "Wld: A robust local image descriptor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1705–1720, 2010.
- [154] Y. Jiang, J. Yuan, and G. Yu, "Randomized spatial partition for scene recognition," in *European Conference on Computer Vision*, pp. 730–743, 2012.
- [155] L. Li and L. Fei-Fei, "What, where and who? classifying events by scene and object recognition," in *International Conference on Computer Vision*, pp. 1–8, 2007.
- [156] P. Saisan, G. Doretto, Y. Wu, and S. Soatto, "Dynamic texture recognition," in *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 58–63, 2001.
- [157] A. Chan and N. Vasconcelos, "Probabilistic kernels for the classification of autoregressive visual processes," in *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 846–851, 2005.
- [158] A. Ravichandran, R. Chaudhry, and R. Vidal, "View-invariant dynamic texture recognition using a bag of dynamical systems," in *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1651–1657, 2009.
- [159] K. G. Derpanis and R. P. Wildes, "Dynamic texture recognition based on distributions of spacetime oriented structure," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 191–198, IEEE, 2010.
- [160] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, Society for Industrial and Applied Mathematics, 2007.

- [161] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," in *Proceeding of International Conference on Analysis and modeling of faces and gestures*, pp. 168–182, Springer-Verlag, 2007.
- [162] A. M. Ferman, A. M. Tekalp, and R. Mehrotra, "Robust color histogram descriptors for video segment retrieval and identification," *Image Processing, IEEE Transactions on*, vol. 11, no. 5, pp. 497–508, 2002.
- [163] P. Phillips, H. Moon, S. Rizvi, and P. Rauss, "The feret evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.
- [164] R. Péteri, S. Fazekas, and M. J. Huiskes, "DynTex : a Comprehensive Database of Dynamic Textures," *Pattern Recognition Letters*, vol. doi: 10.1016/j.patrec.2010.05.009. <http://projects.cwi.nl/dyntex/>.
- [165] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.
- [166] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 524–531, 2005.
- [167] J. Wu and J. Rehg, "Where am i: Place instance and category recognition using spatial pact," in *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.
- [168] J. Wu and J. Rehg, "Centrist: A visual descriptor for scene categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1489–1501, 2011.
- [169] C. M. Jarque and A. K. Bera, "A test for normality of observations and regression residuals," *International Statistical Review*, pp. 163–172, 1987.
- [170] R. B. D'agostino, A. Belanger, and R. B. D'Agostino Jr, "A suggestion for using powerful and informative tests of normality," *The American Statistician*, vol. 44, no. 4, pp. 316–321, 1990.
- [171] H. C. Thode, *Testing for normality*. CRC Press, 2002.
- [172] G. W. Corder and D. I. Foreman, *Nonparametric statistics for non-statisticians: a step-by-step approach*. John Wiley & Sons, 2009.

- [173] C. Flecher, D. Allard, and P. Naveau, "Truncated skew-normal distributions: moments, estimation by weighted moments and application to climatic data," *Metron*, vol. 68, no. 3, pp. 331–345, 2010.
- [174] J. Wu, "Power mean SVM for large scale visual classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2344–2351, IEEE, 2012.
- [175] J. Wu, Y. Zhang, and W. Lin, "Towards good practices for action video encoding," in *Computer Vision and Pattern Recognition (CVPR)*, pp. 2577–2584, June 2014.
- [176] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang, "Probabilistic elastic matching for pose variant face verification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3499–3506, IEEE, 2013.
- [177] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher vector faces in the wild," in *British Machine Vision Conference*, vol. 1, p. 7, 2013.
- [178] K. G. Derpanis and R. P. Wildes, "Spacetime texture representation and recognition based on a spatiotemporal orientation analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 6, pp. 1193–1205, 2012.