Column Flotation of Subbituminous Coal Using the Blend of Trimethyl Pentanediol Derivatives and Pico-Nano Bubbles

Jinxiang Chen

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Recommended Citation
https://researchrepository.wvu.edu/etd/5341
Patch-based Ensemble Learning Scheme for Heterogeneous Face Recognition

by

Cunjian Chen

Doctoral Dissertation submitted to the College of Engineering and Mineral Resources at West Virginia University in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

Arun Ross, Ph.D., Chair
Donald Adjeroh, Ph.D.
Guodong Guo, Ph.D.
Xin Li, Ph.D.
Cunquan Zhang, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2014

Keywords: Face Recognition, Facial Cosmetics, Gender Estimation, Age Estimation, Multispectral Imagery, Cross-spectral Matching, Near-infrared, Thermal

Copyright 2014 Cunjian Chen
Abstract

Patch-based Ensemble Learning Scheme for Heterogeneous Face Recognition
by
Cunjian Chen
Doctor of Philosophy in Computer Science
West Virginia University
Arun Ross, Ph.D., Chair

The problem of Heterogeneous Face Recognition (HFR) involves comparing and matching face images that look significantly different primarily due to variations in their photometric composition. Examples include matching face images acquired in different spectral bands (e.g., visible versus thermal spectrum), or before and after the application of makeup. In this dissertation, we develop and evaluate a robust face recognition method to address this challenge.

The first part of the thesis deals with the topic of facial cosmetics. In this regard, we demonstrate the negative impact of facial cosmetics on existing face recognition as well as gender and age estimation systems. Next, we design a method that automatically detects makeup in face images. The proposed method extracts a feature vector that captures the shape, texture and color characteristics of the input face image, and employs a learning approach based on SVM/AdaBoost to determine the presence or absence of makeup. Finally, we design a patch-based ensemble learning method to perform makeup-invariant face recognition. In the proposed scheme, each face image is tessellated into patches and each patch is represented by a set of feature descriptors, viz., Local Gradient Gabor Pattern (LGGP), Histogram of Gabor Ordinal Ratio Measures (HGORM) and Densely Sampled Local Binary Pattern (DS-LBP). Then, a novel Semi Random Subspace Linear Discriminant Analysis (SRS-LDA) method is used to perform ensemble learning by sampling patches and constructing multiple common subspaces between before-makeup and after-makeup facial images. Finally, collaborative-based and sparse-based representation classifiers are used to compare feature vectors in this subspace and the resulting scores are combined via the sum-rule. Extensive experimental analysis demonstrates the efficacy of the proposed method.

The second part of the thesis deals with the topic of cross-spectral face recognition. Here, we design a method to compare input face images originating from the visible (VIS) and thermal (THM) spectrum. In the training phase of the proposed method, face images from VIS and THM are filtered and tessellated into patches. Each patch is represented using Pyramid Scale Invariant Feature Transform (PSIFT) or Histograms of Principal Oriented Gradients (HPOG). Then, a cascaded subspace learning process, consisting of whitening transformation, factor analysis, and common discriminant analysis, is used to construct multiple common subspaces between VIS and THM facial images. During the matching phase, the projected feature vectors from individual subspaces are concatenated to form a single feature vector. A Nearest Neighbor (NN) classifier is then used to compare feature
vectors and the resulting scores corresponding to three image filters are combined via the sum-rule. The proposed face matching algorithm is evaluated on two multispectral face datasets and is shown to achieve very good results.

In summary, the primary contribution of this dissertation is the design of a novel patch-based ensemble learning method in conjunction with a cascaded subspace learning process to perform effective heterogeneous face recognition.
Acknowledgments

I would first like to thank my committee chair and advisor, Dr. Arun Ross, for giving me the opportunity to work with him. This dissertation would not be possible without his constant guidance and steadfast support. His passion, vision, and love for research is always an inspiration source and influence to me. He has spent huge amount of effort in editing and shaping each submitted manuscript. I am always amazed that even a small error in writing can also be quickly pointed out. I deeply admire him for his attention to details. His attitude towards effective scientific writing has significantly impacted my research. Dr. Ross is the most hard-working professor I have ever seen who dedicates most of his time in doing research and publishing high quality papers. We can always find him in the office and he is very pleased to help you with any research questions. During my PhD career, Dr. Ross has provided me many valuable opportunities to attend research conferences to help me build the network and expand the vision.

Dr. Ross is also an outstanding professor in teaching, from which I have benefited a lot in his classes. His classes are always very well prepared. He is very talented in dividing a complex problem into subproblems and explain it in a very clear way to his students. As a mentor, Dr. Ross also cares about students’ life. When I feel dismayed about my future, he will utilize any available resources, provide encouragement and motivations for me to continue the career. I am forever indebted to him for helping me complete the PhD journey and shaping me as a researcher.

I would also like to thank Dr. Donald Adjereh, Dr. Guodong Guo, Dr. Xin Li and Dr. Cunquan Zhang for being on my committee. I truly appreciate to have you evaluate my research work and provide valuable comments. Your insightful thoughts and ideas on my work has helped improve the quality of this thesis.

I am very fortunate to work with Dr. Antitza Dantcheva. She is not only a close collaborator in my research, but also a good friend in my life. We have spent numerous times
discussing and exchanging ideas about makeup-related research work. She is always ready to offer assistance when I encounter difficulties. I remember that Dr. Dantcheva devoted much of her time in helping me with the thesis proposal. Besides, she is very optimistic and always encourages me in my studies and life.

In my daily work I have been blessed with a friendly and cheerful group of fellow students. My colleagues in the lab, such as Dr. Simona Crihalmeanu, Yaohui Ding, Dr. Asem Othman, Raghunandan Pasula, Dr. Ajita Rattani and Thomas Swearingen who provided useful suggestions on the thesis work or the discussion in relevant work. I am also grateful to Dr. Hu Han, with whom I have learned a lot in solving face recognition research problems. He is also a good friend in my personal life.

Last but not the least, I wish to thank my parents, my brother, and all other relatives. They raised me, supported me, taught me, and loved me. Living a thousand miles away, my mother is a person who always takes good care of household. Her undivided attention and unwavering love has empowered me to finish the path. To them I dedicate this thesis. I also want to thank other people who are part of my life and have helped me transit in USA. They made my stay in the USA wonderful for both, to study and live.
# Contents

Acknowledgments v
List of Figures ix
List of Tables xvi
Notation xix

1 Introduction 1
   1.1 Face Recognition ................................. 4
      1.1.1 Categorization of Face Recognition .......... 5
      1.1.2 Challenges of Face Recognition .......... 7
   1.2 Challenges Due to Makeup ......................... 12
      1.2.1 Impact on Automated Face Recognition Algorithms .... 15
      1.2.2 Impact on Gender Estimation Algorithm .... 16
      1.2.3 Impact on Age Estimation Algorithm .... 17
   1.3 Cross-spectral Matching Challenge ............. 18
   1.4 Thesis Contributions ........................... 20
   1.5 Thesis Organization ........................... 22

2 Impact of Cosmetics on Face Recognition 25
   2.1 Facial Makeup Databases ......................... 27
      2.1.1 YouTube Makeup (YMU) Database ........... 27
      2.1.2 Virtual Makeup (VMU) Database .......... 28
   2.2 Face Recognition Techniques ..................... 29
      2.2.1 Gabor Wavelets ............................ 30
      2.2.2 Local Binary Pattern (LBP) ............. 32
      2.2.3 Neurotechnology Verilook ............... 33
   2.3 Experiments ..................................... 33
      2.3.1 Experiments on the YMU Database ........ 33
      2.3.2 Experiments on the VMU Database .......... 35
   2.4 Observations .................................... 38
   2.5 Addressing the Problem ......................... 38
   2.6 Influence of Demographics ..................... 40
   2.7 Summary ....................................... 42
## Contents

### 3 Impact of Cosmetics on Gender and Age

3.1 Makeup Induced Gender Alteration .................................. 49
   3.1.1 Makeup Induced Gender Alteration (MIGA) Dataset ........... 50
   3.1.2 Gender Classification and Alteration Metrics ............... 52
   3.1.3 Gender Estimation Algorithms ............................... 53
   3.1.4 Gender Spoofing Experiments ............................... 55
3.2 Makeup Induced Age Alteration .................................... 57
   3.2.1 Description of Datasets .................................... 58
   3.2.2 Age Estimation and Alteration Metrics ....................... 59
   3.2.3 Age Estimation Algorithm .................................. 59
   3.2.4 Age Alteration Experiments ............................... 60
3.3 Discussion ..................................................... 61
3.4 Summary ...................................................... 63

### 4 Automatic Facial Makeup Detection

4.1 Makeup Detection ................................................ 66
   4.1.1 Colorspace ............................................... 68
   4.1.2 Proposed Method ........................................... 68
   4.1.3 ROI Detection .............................................. 69
   4.1.4 Feature Extraction ....................................... 71
4.2 Classification .................................................. 74
4.3 Makeup Databases .............................................. 75
   4.3.1 YouTube Makeup Database (YMU) .......................... 75
   4.3.2 Makeup in the Wild Database (MIW) ....................... 76
4.4 Experiments .................................................... 76
   4.4.1 Makeup Detection ........................................... 77
4.5 Application in Face Recognition .................................. 79
4.6 Summary ...................................................... 81

### 5 Makeup-Robust Face Recognition

5.1 Motivation and Related work .................................... 83
5.2 Proposed Method ................................................ 84
5.3 Feature Descriptors ............................................ 86
   5.3.1 Local Gabor Gradient Pattern (LGGP) .................... 88
   5.3.2 Histogram of Gabor Ordinal Ratio Measures (HGORM) .... 88
   5.3.3 Densely Sampled Local Binary Pattern (DS-LBP) ........ 90
5.4 Semi-random Subspace LDA (SRS-LDA) .......................... 91
   5.4.1 Training Phase ............................................ 92
   5.4.2 Testing Phase ............................................. 95
5.5 Baseline Algorithms ........................................... 96
5.6 Experiments ................................................... 98
   5.6.1 Makeup Datasets .......................................... 99
   5.6.2 Experiment on the YMU Database ......................... 100
   5.6.3 Large-scale Identification Experiment .................... 107
   5.6.4 Comparison against HFR Matcher ......................... 110
## Contents

5.6.5 Experiment on the FERET Database .............................................. 110  
5.7 Discussions ............................................................................. 111  
5.8 Summary .................................................................................. 112  

6 Cross-spectral Face Recognition ....................................................... 114  
6.1 Local Gabor Gradient Pattern (LGGP) ........................................ 114  
6.1.1 Image Preprocessing .............................................................. 116  
6.1.2 Descriptors and Encoding Methods ........................................ 118  
6.1.3 Our Proposed Technique ....................................................... 119  
6.1.4 Evaluation on FERET Benchmark Datasets ......................... 121  
6.1.5 Evaluation on HFB Dataset .................................................... 125  
6.2 Matching Remote Near-infrared to Visible .................................. 129  
6.2.1 Face Descriptors .................................................................... 131  
6.2.2 Random Subspace Linear Discriminant Analysis (RS-LDA) .... 134  
6.2.3 Whitening Transformation ..................................................... 136  
6.2.4 Database .............................................................................. 137  
6.2.5 Intra-spectral and Cross-distance Face Matching ................... 137  
6.2.6 Cross-spectral and Cross-distance Face Matching .................. 138  
6.3 Matching Thermal to Visible ....................................................... 143  
6.3.1 Proposed Matching Framework ............................................ 146  
6.3.2 Factor Analysis Model ........................................................... 149  
6.3.3 Baseline Algorithms ............................................................. 152  
6.3.4 Evaluation on PCSO dataset .................................................. 154  
6.3.5 Identity Component Reconstruction ...................................... 158  
6.3.6 Evaluation on Carl Dataset .................................................... 159  
6.4 Summary ................................................................................... 162  

7 Summary and Future Work ................................................................ 164  
7.1 Summary ................................................................................... 164  
7.2 Future Work ............................................................................. 166  

8 Appendix ....................................................................................... 168  
8.1 List Of Publications ................................................................. 168  

References ....................................................................................... 170
List of Figures

1.1 Examples of some of the human traits that have been used for biometric recognition. Iris image is from CASIA-Iris Database (http://www.idealtest.org/); Face image is from FERET Database [1]; Hand geometry, voice and signature images are from [2].

1.2 The two modes of recognition: verification and identification. Face images are from FERET database. The numbers represent the similarity match scores between samples as computed by a commercial biometric system.

1.3 The categorization of face recognition techniques. The number is used to indicate the time when the method was first published. Only a few representative methods are listed here.

1.4 A typical face recognition system consists of four different modules. Face image is from FERET Database.

1.5 Illustration of different HFR scenarios. Face images of (a) are from CUHK sketch database. Face images of (b) are from HFB database. Face images of (c) are from FGNET database. Face images of (d) are from Carl multispectral database. Face images of (e) are from YouTube. The matching scenarios of (d) and (e) are considered in this dissertation.

1.6 Pose, illumination and expression (PIE) challenges. A face image (a) with variations in facial pose (b), illumination (c), and expression (d). All the face images are obtained from the FERET database. The numbers represent the similarity match scores between (a) and each sample image (including sample (a)) as computed by a commercial face recognition system. Match score “0” means the matcher fails to provide a similarity match score.

1.7 Challenges in face recognition: Face images of actress Jennifer Grey demonstrating variations due to pose, illumination, expression, as well as alterations due to plastic surgery, aging and makeup. Images obtained from www.imdb.com, www.nydailynews.com and YouTube. The numbers represent the similarity match scores between (a) and each sample image (including sample (a)) as computed by a commercial face recognition system.

1.8 Makeup product market size in Europe, United States, Japan and China. The figure is generated based on the statistical data in report [3].

1.9 Different makeup products; Images are from http://www.sephora.com.
LIST OF FIGURES

1.10 An example of showing how makeup can be used to change the overall facial appearance, resulting in a possible false non-matching case. Image is obtained from the World Wide Web. ................................................................. 15

1.11 The effect of makeup. The subject in (a) is shown after the application of makeup in (b) - (f). Images are obtained from YouTube. ................................. 15

1.12 Examples of a male subject before and after the application of makeup. From left to right, makeup varies from light to heavy. Images are obtained from YouTube. ................................................................. 16

1.13 The two sets (rows) of images correspond to two different race groups. First column are images without makeup and the last two columns are images with makeup. Images are obtained from YouTube. ................................. 17

1.14 An example of showing how makeup can change the perception of gender, resulting in a possible gender estimation error. Images are obtained from YouTube. ................................................................. 18

1.15 An illustration of different spectral bands. Visible, near-infrared and midwave infrared spectral bands are studied in this work. ................................. 19

2.1 Examples of two subjects in the YMU database assembled by the authors, each without [(a) and (c)] and with makeup [(b) and (d)]. The subject on the top has subtle makeup, while the one on the bottom has heavy makeup. The output of the automated face and eye detection scheme, see [4], is indicated in yellow. ................................................................. 28

2.2 Example of a subject in the VMU database assembled by the authors. (a) depicts the subject without makeup, whereas, (b), (c) and (d) constitute the makeup shots. The makeup shots include the synthetic addition of (b) eye makeup (c) lipstick, and (d) full makeup, respectively, using the popular Taaz software. ................................................................. 29

2.3 (a) Example makeup ($M$) image from the YMU database. (b) Preprocessed image. (c) Gabor magnitude response (40 images). (d) LBP coded image. We note that the Gabor response around the mouth and ocular regions is more prominent than in other parts of the face. ................................................................. 31

2.4 Sample subject from the YMU database. Intra-class variations can be observed between the no-makeup shots (a) and (b) as well as the makeup shots (c) and (d). In (e), the distance scores as reported by the three face matchers are presented for the various matching scenarios. ................................................................. 35

2.5 Boxplot of the genuine score distributions for the YMU database as computed using the LBP face matcher for the 3 matching scenarios: $M_1$ vs $M_2$, $M$ vs $N$, $N_1$ vs $N_2$. ................................................................. 36

2.6 Example of normalized and cropped images for a subject in the VMU database. (a) No makeup. (b) Eye makeup. (c) Lipstick only. (d) Full makeup. In (e), the distance scores as reported by the three face matchers are presented for the various matching scenarios. ................................................................. 37

2.7 Boxplot of the genuine score distributions as computed by the LBP face matcher on the VMU database for the 3 matching scenarios: $N$ vs $E$, $N$ vs $L$ and $N$ vs $F$. ................................................................. 37
2.8 LGBP matcher. (a) Preprocessed sample makeup image from the YMU database. (b) The 40 LBP maps generated after applying LBP to 40 different Gabor magnitude responses.

2.9 Performance of the four face matchers on the YMU database. (a) Gabor; (b) LBP; (c) Verilook; (d) LGBP. We observe consistent performance decrease in the case $M$ vs $N$ for all 4 algorithms. LGBP provides the best results with an EER of 15.89%, this error rate suggests that makeup remains a challenge and requires more research.

2.10 Performance of the three face matchers on the Asian dataset. (a) LGBP (16.30% EER); (b) LGGP (16.38% EER); (c) HMBP (15.84% EER). We observe consistent performance decrease in the case $M$ vs $N$ for all 3 algorithms.

2.11 Performance of the three face matchers on the African dataset. (a) LGBP (17.54% EER); (b) LGGP (16.08% EER); (c) HMBP (17.89% EER). We observe consistent performance decrease in the case $M$ vs $N$ for all 3 algorithms.

2.12 Performance of the three face matchers on the Male dataset. (a) LGBP (7.63% EER); (b) LGGP (8.06% EER); (c) HMBP (6.69% EER). We observe consistent performance decrease in the case $M$ vs $N$ for all 3 algorithms.

3.1 Examples of subjects applying facial makeup for gender spoofing (from YouTube). Male-to-female (a-b): foundation conceals facial hair and skin blemishes; eye and lip makeup are then applied in the way females usually do. Female-to-male (c-d): dark eye-shadow is used to contour the face shape and the nose; then, thicker eye-brows, mustache, and beard are simulated using special makeup products. Only the facial region is used in this study (see Figure 3.3).

3.2 Example images from the Makeup Induced Gender Alteration (MIGA) dataset: (a) male-to-female subset: male subjects apply makeup to look like females, and (b) female-to-male subset: female subjects apply makeup to look like males. In both (a) and (b), the images in the upper row are before makeup and the ones below are the corresponding images after makeup.

3.3 The example images from Figure 3.2 after preprocessing.

3.4 The output of the COTS gender classifier on the images shown in Figure 3.2. M indicates “classified as male” and F indicates “classified as female”. While all male-to-female transformations were successful in this example, only the leftmost and rightmost female-to-male transformations were successful.

3.5 Example image pairs from the MIAA dataset. Top row: images before makeup; Bottom row: corresponding images after makeup.

3.6 MAD results of the OpenBR age estimation algorithm on the (a) YMU and (b) VMU datasets. (a) We observe a higher MAD value for the A vs B case; (b) We observe higher MAD values for eye makeup and full makeup.

3.7 Age defying effect of makeup on the OpenBR algorithm. x-axis values indicate the difference in estimated age in years (after makeup - before makeup), while y-axis values show the percentage of subjects in MIAA.

3.8 Automatic age estimation results before and after the application of makeup (original images are shown in Figure 3.5). Top row: images before makeup; Bottom row: corresponding images after makeup.
4.1 The before and after makeup images of two subjects. The after makeup images exhibit color, shape and texture changes in the eye and mouth regions. 

4.2 Visualizing a face as individual channels in the RGB (top) and HSV (bottom) colorspaces. 

4.3 Proposed framework for automatic facial makeup detection. 


4.5 The tessellation scheme used for extracting color-based features. 

4.6 Examples of edge orientation histograms for the same subject with and without makeup. The application of makeup typically increases edge information around the eyes and the mouth. 

4.7 Facial images showing variations in pose, illumination, expression and resolution from the YMU database [5]. The corresponding eye and mouth ROIs are shown in Fig. 4.4. 

4.8 Sample images from the MIW database. Images are collected from the Internet. Top row shows images without makeup and the bottom row shows images with makeup. Note the unconstrained nature of the images. 

4.9 ROC curves of SVM-based and Adaboost-based makeup detectors on the YMU database for all 5 trials in the cross-validation scheme [(a) - (e)] and on the MIW database [(f)]. The parameters used by the SVM is indicated below each graph. The numbers within parentheses in each legend indicate the AUC values. 

4.10 Output of the proposed makeup detector on the MIW images shown in Fig. 4.8. The detector was trained using all the images from the YMU database [5]. 

5.1 Proposed framework for matching after-makeup images with before-makeup images. During the training phase, for each feature descriptor, a pool of patches are extracted, followed by weight learning, patch sampling and random subspace construction. In the testing phase, patches from an input image are projected onto the learned random subspace. A combination of SRC and CRC classifiers are used to compare feature vectors in these subspaces and generate a match score. This process is repeated for each descriptor and the matching scores corresponding to individual feature descriptors are fused to generate the final similarity score. 

5.2 Illustration of ordinal filters at different distances and orientations (horizontal, vertical). Positive and negative lobes are arranged in various configurations in terms of number of lobes and orientations between lobes. 

5.3 The horizontal and vertical ordinal filters used to compute the ratio measure. The distances between positive and negative lobes are same for these two filters.
5.4 An example showing how SRC is used for identity classification during the ensemble learning. CRC follows the same principle, except that coefficients are spread widely. $\rho^{(k)}$ denotes the coefficients generated for the $k$-th subspace of a particular descriptor. Horizontal axis refers to the number of gallery samples and vertical axis refers to the coefficients for each sample as computed in Eqn. (6.29). The red circle denotes the coefficient associated with the first gallery sample in each subspace. The identity of a probe sample is correctly inferred after summing all the coefficients in the red circle. 97

5.5 The example images after alignment and cropping. Substantial change in facial appearance is observed after the application of makeup. 99

5.6 Sample images from the T-makeup training dataset. Top row shows images before the application of makeup and bottom row shows images after the application of makeup. 100

5.7 Performance evaluation of proposed method and different COTS systems with reported ROC curves (EER: A vs. B) on YMU database. 102

5.8 Performance evaluation of proposed method and different COTS systems with reported ROC curves (GARs at different FAR: A vs. B) on YMU database. This is the semilogarithmic plot of Figure 5.7. 103

5.9 Demonstration (boxplot) of stability of the SRS-LDA method on YMU database: 7.52 ± 0.23 (EER), 68.51 ± 0.59 (GAR at FAR=0.1%). 104

5.10 The impact of number of subspaces on the SRS-LDA algorithm for different feature descriptors (YMU database). 106

5.11 The CMC curves of different face matchers before and after adding mugshots. YMU only: 302 after-makeup samples from YMU are used as probes and 302 before-makeup samples from YMU are used in the gallery; YMU + FRGC Mugshots: 302 after-makeup samples from YMU are used as probes, while 302 before-makeup samples from YMU and 10K FRGC mugshots are used in the gallery; YMU + FRGC Mugshots + MIW: 302 after-makeup samples from YMU are used as probes, while 302 before-makeup samples from YMU, 112 after-makeup samples from MIW [6], and 10K mugshots from FRGC are used in the gallery. 108

5.12 Reported Rank-k identification rates on FERET using the proposed method. The T-makeup dataset is used to train the SRS-LDA face matcher. 111

6.1 The general architecture considered in this work. An input image is subject to a Gabor Filter resulting in a Gabor Response (magnitude here). The Gabor Magnitude is encoded using different descriptors. The resulting coding values are represented as spatial histograms. The image is from the FERET database. 115

6.2 Examples of WLD and Gradient coding schemes. The original images are of the same subject from the FERET dataset under different illumination conditions. 122
LIST OF FIGURES

6.3 An example showing how to compute the Gradient descriptor. \( x_c \) is the center pixel surrounded by neighbors equally sampled from \( x_0 \) to \( x_{P-1} \), where \( P \) is the neighborhood size. \( r \) is the defined radius of the neighborhood. Different parameters such as scale, orientation, and neighborhood size can be selected accordingly. The two given equations compute the descriptor at different orientations of 0° and 45°, respectively. 

6.4 Performance evaluation of different methods with the reported CMC curves.

6.5 Samples of face images in visible (top) and near-infrared (bottom) spectra. Images are from HFB database and not co-registered.

6.6 Performance evaluation of cross-spectral matching on HFB database. VIS images are used as Gallery, while NIR images are used as Probe.

6.7 An NIR probe image acquired at a distance of 150 meter (150m) is matched against a VIS gallery image at a distance of 1 meter (1m). The quality of NIR image is severely degraded by sensor noise, blur and other covariates.

6.8 Exemplar images of the same subject from LDHF database after the process of image filtering. Top row shows the results after applying image filters to a visible image at 1m distance; Bottom row shows the results after applying image filters to a near-infrared image at 150m distance.

6.9 An illustration of Pyramid-SIFT feature extraction at five different image resolutions. The blurring effect increases with the downsampling factor. The number of patches extracted from each image resolution is 108, 48, 24, 12 and 4. For visualization purpose, the image is upscaled to the original resolution.

6.10 The example images of LDHF-DB after alignment and cropping. Each row denotes a subject that is captured at 1m VIS and 1m NIR, 60m NIR, 100m NIR and 150m NIR.

6.11 Rank-1 performance evaluation of PSIFT descriptor with different image filters on LDHF-DB. Intra-spectral and cross-distance face matching is shown here.

6.12 Rank-1 performance evaluation of PSIFT descriptor with different image filters on LDHF-DB. Cross-spectral and cross-distance face matching is considered here.

6.13 Reported ROC curve for the cross-spectral and cross-distance face recognition for the proposed method (without whitening).

6.14 A decrease in similarity score is observed when matching a THM image to a VIS image. The range of similarity scores is between 0 and 1.

6.15 The proposed framework for matching thermal to visible face images. A cascaded subspace learning process (whitening transformation, factor analysis and common discriminant analysis) involving both VIS and THM images is used during the training phase. The Gallery and Probe samples are then projected onto the learned subspaces during the testing phase.

6.16 Exemplar images of the same subject from the PCSO database after image filtering. Top row shows the results after applying image filters to a VIS image; Bottom row shows the results after applying image filters to a THM image.
6.17 Samples of face images in visible (top) and thermal (bottom) spectra. Images are from PCSO database and not co-registered. .................................................. 154
6.18 ROC and CMC curves of the proposed matching method on the PCSO dataset for PSIFT and HPOG descriptors .......................................................... 156
6.19 Illustration of generating spectral-invariant identity images across different spectra on PCSO test set. VIS and THM samples from the same subject are used here. These results are based on the learning of HFA model on PCSO training dataset. For each subject, top row shows the original VIS and THM images and bottom row shows the corresponding reconstructed spectral-invariant identity images. .................................................. 159
6.20 Samples of face images in VIS (top) and THM (bottom) spectra. Images are from Carl database and not co-registered. .................................................. 160
6.21 ROC and CMC curves of different face matchers on Carl dataset for PSIFT and HPOG descriptors. .......................................................... 161
List of Tables

1.1 Challenges in facial cosmetics. ................................. 10
1.2 Challenges in multispectral imagery. .......................... 10
1.3 Examples of face altering makeup items. ......................... 13
1.4 Illustration of face image samples captured across different spectra and distances. Images are from Long Distance Heterogeneous Face Database (LDHF-DB) [7]. ................................................. 21
1.5 Illustration of face image samples captured under visible and thermal (MWIR: 3-5µm) spectra. Images are from Pinellas County Sheriff’s Office (PCSO) Thermal Face Dataset. ........................................... 22

2.1 EER (%) of the face matchers on the YMU database. In all cases, comparing a makeup image (M) against a no-makeup image (N) decreases accuracy. 34
2.2 EER (%) of the face matchers on the VMU database. Matching a no-makeup image (N) against a full makeup image (F) results in decreased performance. 36
2.3 EER (%) of LGBP matcher on the YMU database. .................. 40
2.4 EER (%) of LGBP matcher on the VMU database. .................. 40
2.5 Equal Error Rates (%) corresponding to the seven face matchers on an extended version of YMU dataset with 151 subjects and 604 images. ............ 42

3.1 Male classification rates (%) and GSI values (%) corresponding to the three gender classification algorithms on the male-to-female subset of the MIGA dataset. ......................................................... 56
3.2 Female classification rates (%) and GSI values (%) corresponding to the three gender recognition algorithms on the female-to-male subset of the MIGA dataset. 57

4.1 Examples of face altering makeup items. ......................... 67
4.2 The dimensionality of features used in makeup detection. ....... 74
4.3 The classification rates of the SVM-based and Adaboost-based makeup detector on the YMU database. ........................................ 79
4.4 Classification Rates of the SVM and Adaboost classifiers on the YMU database. The numbers in parentheses indicate the number of “no-makeup” and “makeup” images in each trial. ................................. 79
4.5 Face verification performance (%) at a FAR of 1% before and after \((B/A)\), applying the proposed face pre-processing scheme. The pre-processing scheme is invoked only when one image is deemed to have makeup and the other image is deemed to be without makeup by the proposed makeup detector. Column 3 highlights the improvement in verification results for the \(M\) vs \(N\) case.  

5.1 Summary of patch-based feature descriptors used in the proposed framework.  

5.2 Equal Error Rates (%) corresponding to the eight face matchers and three matching scenarios \((B \text{ vs. } B:\) matching of before-makeup images, \(A \text{ vs. } A:\) matching of after-makeup images, and \(A \text{ vs. } B:\) one of the images is after-makeup, the other is before-makeup) on YMU database (151 subjects and 604 images).  

5.3 The result of fusing the proposed method with different COTS systems on the YMU database. GAR performance at a FAR of 0.1% is reported.  

5.4 Performance of individual feature descriptors on YMU database. GAR performance at a FAR of 0.1% is reported.  

5.5 The computational time (seconds) for the proposed algorithm in terms of feature extraction and classification steps.  

5.6 Rank-1 accuracies (%) of different face matchers before and after adding FRGC mugshots to the gallery.  

5.7 Examples of true and false matches at Rank-1. The selection of these matching pairs \((A \text{ vs. } B)\) is based on the face identification scenario.  

5.8 Examples of successful Rank-1 matches that demonstrate the effectiveness of fusion (Proposed+COTS-2+COTS-3).  

5.9 Comparison against a HFR matcher [8]. The proposed method uses both SRC and CRC classifiers. GAR performance at a FAR of 0.1% is reported.  

5.10 Rank-1 performance (%): comparison of the proposed method with other state-of-the-art commercial matchers on the FERET dataset.  

6.1 Rank-1 performance (%): comparison of different coding schemes against the baseline (Base).  

6.2 Rank-1 performance (%): comparison of the proposed methods with other state-of-the-art descriptor-based methods on FERET.  

6.3 Evaluation of CSDN image filter on cross-spectral and cross-distance face matching. GAR (%) performances at both FARs of 0.1% and 1% are reported, along with Rank-1 accuracy (%).  

6.4 Evaluation of GIST image filter on cross-spectral and cross-distance face matching. GAR (%) performances at both FARs of 0.1% and 1% are reported, along with Rank-1 accuracy (%).  

6.5 Evaluation of SQI image filter on cross-spectral and cross-distance face matching. GAR (%) performances at both FARs of 0.1% and 1% are reported, along with Rank-1 accuracy (%).  

6.6 CSDN+GIST+SQI: GAR performances at both FARs of 0.1% and 1% are reported, along with Rank-1 accuracy.
6.7 CSDN+GIST+SQI (Whitening): GAR performances at both FARs of 0.1% and 1% are reported, along with Rank-1 accuracy. ......................... 142
6.8 Comparison against the HFR matchers (GSIFT and GSIFT+COTS) in [9]. The proposed method with or without whitening transformation is reported in the last column. GAR (%) performances at 1% FAR are reported. .... 143
6.9 Evaluation of different face matchers on PCSO thermal dataset with PSIFT descriptor. GAR (%) performances at both 0.1% FAR and 1% FAR are reported, along with Rank-1 accuracy (%). ................................. 155
6.10 Evaluation of different face matchers on PCSO thermal dataset with HPOG descriptor. GAR (%) performances at both 0.1% FAR and 1% FAR are reported, along with Rank-1 accuracy (%). ................................. 155
6.11 Evaluation of the proposed RSLDA+Whitening+HFA (PSIFT) method on PCSO thermal dataset with different image filters. GAR (%) performances at both 0.1% FAR and 1% FAR are reported, along with Rank-1 accuracy (%). 156
6.12 Comparison of the proposed matcher against state-of-the-art matchers on PCSO thermal dataset. The proposed method is the score-level fusion of PSIFT and PHOG with RSLDA+Whitening+HFA. GAR (%) performances at both 0.1% FAR and 1% FAR are reported, along with Rank-1 accuracy (%). 157
6.13 Evaluation of different face matchers on Carl thermal dataset with PSIFT descriptor. GAR (%) performances at both 0.1% FAR and 1% FAR are reported, along with Rank-1 accuracy (%). ................................. 160
6.14 Evaluation of different face matchers on Carl thermal dataset with HPOG descriptor. GAR (%) performances at both 0.1% FAR and 1% FAR are reported, along with Rank-1 accuracy (%). ................................. 162
6.15 Comparison of the proposed matcher against state-of-the-art matchers on Carl thermal dataset. The proposed method is the score-level fusion of PSIFT and PHOG with RSLDA+Whitening+HFA. GAR (%) performances at both 0.1% FAR and 1% FAR are reported, along with Rank-1 accuracy (%). .... 162
Notation

We use the following notation and symbols throughout this thesis.

\( x_i \) : Vector representation of an image \( i \)
\( \hat{x} \) : Mean vector representation
\( (\cdot)^T \) : Matrix transpose
\( \Sigma_g \) : Covariance Matrix
\( \langle \cdot \rangle \) : Dot product of two vectors
\( \text{sign}(\cdot) \) : Sign function
\( \| \cdot \| \) : Euclidian norm
\( \mathbb{R}\{\cdot\} \) : Real part of the argument
\( \mathbb{I}\{\cdot\} \) : Imaginary part of the argument
\( \mathcal{N} \) : Gaussian Distribution

Bold upper case letters denote matrices and bold lower case letters denote vectors.
Chapter 1

Introduction

Biometrics is the science of recognizing individuals based on physical or behavioral traits, such as face, fingerprint, iris, hand geometry or voice [10, 2] (see Figure 1.1). Biometric technology has been adopted in state-of-the-art intrusion detection and security systems [2]. A biometric system automatically captures one or more physical or behavioral characteristics from an individual for recognition purposes. Distinct features, which are extracted from a captured biometric sample (e.g., face image, fingerprint image, etc.), are stored as a template in the system. A collection of these templates constitutes a biometric database. Each template is labeled with an identifier, such as a name or a number, that is used to denote identity. A biometric system consists of various modules: (a) a sensor module (e.g., infrared camera sensor, optical fingerprint sensor, etc.) that is used to collect a biometric sample; (b) a feature extraction module where discriminative features are extracted from the biometric sample; (c) a database module where all the biometric templates are stored along with an identifier; (d) a matcher module where a matching score between two biometric samples is computed by matching their corresponding feature sets.

The working of a biometric recognition system can be described in two stages, namely, enrollment stage and recognition stage. During the enrollment stage, a biometric sample is acquired from an individual via the sensor. Then features are extracted from the sample and stored in the database, along with an identifier denoting the individual. During the recognition stage, a new biometric sample (also referred to as a probe sample) is re-acquired from the individual and matched against the stored templates in the database (also referred
to as gallery samples) to recognize the individual.

There are two modes of recognition in a biometric system: verification and identification. In verification (Figure 1.2(a)), an individual offers a biometric sample along with a claimed identity. In this mode, the system extracts features from the sample and compares the extracted features with only those templates in the database corresponding to the claimed identity. Therefore, this is referred to as 1-to-1 matching. In identification (Figure 1.2(b)), an individual offers a biometric sample without claiming an identity. In this mode, the system extracts features from the sample and compares the extracted features with all the templates stored in the database to determine the identity. Therefore, this is referred to as 1-to-N matching.

![Examples of biometric traits](image)

Figure 1.1: Examples of some of the human traits that have been used for biometric recognition. Iris image is from CASIA-Iris Database (http://www.idealtest.org/); Face image is from FERET Database [1]; Hand geometry, voice and signature images are from [2].
Figure 1.2: The two modes of recognition: verification and identification. Face images are from FERET database. The numbers represent the similarity match scores between samples as computed by a commercial biometric system.
1.1 Face Recognition

The human face is believed to be the principle trait humans use to recognize each other and, hence, has a special role as a biometric trait [2]. It is useful not only for person recognition, but for also revealing ancillary information aiding recognition, such as gender, age and ethnicity, that are referred to as soft biometrics [11]. Face has several noticeable advantages over other biometric traits, which makes it preferable in many practical applications [2]. Firstly, a face image can be captured at a distance and in a covert manner [12]. Secondly, a face image reveals not only the identity, but also emotions of the person (e.g., sadness and happiness) as well as offers soft biometric information (e.g., gender, age, and ethnicity). Thirdly, the human face is the only biometric trait, that is extensively shared on websites and social networks [2]. In the most prominent social network Facebook\textsuperscript{1}, there are over 1.5 billion active Facebook users worldwide, 53\% of them being female and 47\% of them being male, most of them sharing one or many face images with their peers.

Face recognition is a type of biometric recognition that can ascertain an individual’s identity based on patterns (discriminative features) extracted from a face image [13, 2]. Compared to human-based face recognition, one of the key advantages of machine-based face recognition is its capability to handle a large number of face images [13]. Face recognition has been used in applications that range from personal authentication and video surveillance to human-computer interaction. Major web-companies such as Google, Apple and Facebook have utilized face analysis and recognition in order to enhance the features and services of their respective products [14]. For instance, Google has deployed face recognition in photo management software (Picasa). Facebook has enabled automated tagging of photos in its social website [14] using face recognition. Apple has developed iPhoto software in which face recognition is used to help find and name the photos of a person’s friends\textsuperscript{2}. Moreover, the Department of Homeland Security recently launched a new crowd-scanning program, known as Biometric Optical Surveillance System (BOSS)\textsuperscript{3}, and has started to test and evaluate this

\textsuperscript{1}http://zephoria.com/social-media/top-15-valuable-facebook-statistics/
\textsuperscript{2}http://support.apple.com/kb/PH2369
technology in various surveillance applications.

1.1.1 Categorization of Face Recognition

Face recognition technologies have achieved substantial progress over the past two decades. The various face recognition algorithms can be generally categorized into three main approaches [2]:

- **Appearance-based methods:** In these methods, the pixel values within the face are directly used to perform face recognition by considering the face image as a vector of pixel values (raster-scanned and concatenated from raw pixel intensities). Since the dimensionality of the face vector can be prohibitively large, feature reduction is used to project a high-dimensional face vector into a lower-dimensional subspace spanned by a set of basis vectors. Two well-known methods that can be used to compute this subspace are Eigenface, which was proposed by Turk and Pentland [15], and Fisherface, which was proposed by Belhumeur et al. [16].

- **Model-based methods:** A 2D or 3D face model is built to facilitate face matching across pose variations. For example, Elastic Bunch Graph Matching (EBGM) [17] creates a face image model by extracting Gabor jets from face landmark locations and uses that to form a bunch graph. Active appearance model (AAM) [18] is a statistical model of shape and appearance created by learning from a training set of images with manually labeled landmark points.

- **Texture-based methods:** An image texture is characterized by the number and types of its texels (a texel is the fundamental unit of texture) and the spatial organization or layout of its texels [19]. Textures can be classified as being either regular (with repeated texels) or stochastic (without explicit texels) [20]. A face image can be reckoned as an image texture object between these two extremes. Texture can be analyzed in terms of single values (means, variances, etc.), texture units [21], or histograms of filter responses at multiple scales and orientations [22, 21]. Local Binary Pattern (LBP) [23] is a texture-based micro representation, which characterizes micro-
patterns or micro-structures in the face image by binarizing local neighborhoods based on the differences in pixel intensity between the center pixel and neighborhood pixels, and converting the resulting binary string into a decimal value. Scale-invariant feature transform (SIFT) [24] describes an image patch based on the magnitude, orientation, and spatial distribution of the pixel gradients. Local Gabor Binary Pattern Histogram (LGBP) [25] is a representation based on multi-resolution spatial histogram by concatenating the local region histograms of the LBP-encoded Gabor magnitude patterns.

The above methods are summarized in Figure 1.3. In addition to these proposed methods, sparse representation and dictionary learning have recently become popular in face recognition [26, 27].

The pipeline of a face recognition system can be roughly divided into four main modules (see Figure 1.4): face detection, face normalization, feature extraction and face matching.

*Face Detection* determines the size and location of a face from an image [28]. It is a very critical pre-step for automatic face recognition. Most of the current face detection algorithms treat this task as a two-class classification problem (face/non-face) and employ
Face Detection is the process where a detected face is cropped and aligned based on the eye landmarks, and then preprocessed to achieve illumination invariance. Preprocessing methods include Difference of Gaussian (DoG) filter [32], Self Quotient Image (SQI), Retinex Model, to name a few [33].

Feature Extraction is the process where holistic or local features are used to represent the face. Such a feature vector can be derived from a vector of concatenated image pixel values, a calculation of local statistics (e.g., histogram [23]), or can use even more advanced features such as computing the magnitude and phase output of Gabor filter convolutions [25, 34].

Face Matching is the process where two face images are compared and a matching score is computed. A high similarity (or low distance) score indicates a likelihood that the corresponding images are from the same individual. A list of commonly used face comparison methods are k-nearest neighbors (KNN), Support Vector Machine (SVM) [35] and Sparse Representation Classifier (SRC) [36].

1.1.2 Challenges of Face Recognition

A number of face recognition algorithms have demonstrated promising results under well-controlled scenarios, as evidenced by significant reduction of error rates on several public benchmark databases (e.g., FERET [1], AR [37], and FRGC [38]) and featured in a series
of evaluations conducted by National Institute of Standards and Technology (NIST) [39, 40]. However, performance rates dramatically decrease in imperfect real-life environments, such as demonstrated in the labeled faces in the wild (LFW) database [41]. In addition, the performance of face recognition algorithms degrades in the context of heterogeneous recognition, which involves comparing and matching face images that look significantly different primarily due to variations in their photometric composition. Examples (see Figure 1.5) include matching face images acquired in different spectral bands (e.g., visible versus thermal spectrum) or before and after the application of makeup. The key difficulty in matching heterogeneous face images is to account for the appearance changes due to variations in image rendering [42]. This can be accomplished by designing or learning feature vectors that are robust across heterogeneous conditions [43].

![Figure 1.5: Illustration of different HFR scenarios.](image)

- **Pose, Illumination and Expression (PIE):** (1) Pose (Figure 1.6(b)) denotes the position of the 3D face object with respect to the camera, (2) Illumination (Figure 1.6(c)) is the lighting condition of the environment at the time image is captured, and (3) Ex-
Figure 1.6: Pose, illumination and expression (PIE) challenges. A face image (a) with variations in facial pose (b), illumination (c), and expression (d). All the face images are obtained from the FERET database. The numbers represent the similarity match scores between (a) and each sample image (including sample (a)) as computed by a commercial face recognition system. Match score “0” means the matcher fails to provide a similarity match score.

pression conveys a particular emotion condition by changes in positions of muscles in the face (Figure 1.6(d)) [44].

• Aging and Plastic surgery: Facial aging (Figure 1.7(g)), a natural biological change, is a complex biological process that influences both the 3D shape of the face as well as its texture [45]. Plastic surgery (Figure 1.7(f)) [46], a medically induced change, is a process used to correct feature defects or improving the facial appearance.

• Cosmetics: Facial cosmetics (Figure 1.7(h)) have been used by people to enhance their facial appearance. In general, cosmetics involves the application of skin-care creams, powders, lipsticks, colored contact lenses and other types of products. A subset of cosmetics is called “make-up” which refers primarily to products intended to alter the user’s facial appearance

• Multispectral Imagery: face images of the same subject may differ in appearance due to the change in image spectral bands [43]. For examples, images look differ-

4http://en.wikipedia.org/wiki/Cosmetics
Challenges in facial cosmetics.

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Impact of Cosmetics on Face Recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Challenge 1</td>
<td>Impact of Cosmetics on Face Recognition</td>
</tr>
<tr>
<td>Challenge 2</td>
<td>Impact of Cosmetics on Gender Estimation</td>
</tr>
<tr>
<td>Challenge 3</td>
<td>Impact of Cosmetics on Age Estimation</td>
</tr>
</tbody>
</table>

Challenges in multispectral imagery.

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Face Matching between VIS and NIR images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Challenge 1</td>
<td>Face Matching between VIS and NIR images</td>
</tr>
<tr>
<td>Challenge 2</td>
<td>Face Matching between VIS and THM images</td>
</tr>
</tbody>
</table>

ent when captured in visible (VIS), near-infrared (NIR) and thermal (THM) spectra (Figure 1.5(d)).

While there are many related face recognition challenges, we focus on two main challenges in this dissertation: facial cosmetics and multispectral imagery. These two challenges have received relatively limited attention in the face recognition literature. Makeup is widely used and the appearance of a face image can be drastically changed via the use of make-up or some image retouching software such as Photoshop. Different challenges related to cosmetics have been listed in Table 1.1.

The use of near-infrared and thermal sensors in nighttime surveillance has become increasingly popular due to (1) availability of low-cost cameras; (2) their capability to capture images regardless of illumination conditions; and (3) practical applications in criminal and forensic investigations. The use of different image sensors (NIR and THM) results in substantial appearance differences between VIS and NIR samples, and VIS and THM samples. This has become an emerging topic and is beginning to receive more attention from the biometrics community [42]. Different challenges related to multispectral imagery have been listed in Table 1.2.

Therefore, face recognition across cosmetics and multispectral imagery poses significant challenges. Successful solutions to these face recognition problems will greatly expand the capabilities of a face recognition system in a covert scenario (e.g., face recognition in nighttime environments) or in situations where the face image is deliberately altered by the application

[^5]: http://www.wikihow.com/Apply-Makeup-in-Adobe-Photoshop-Cs3
Figure 1.7: Challenges in face recognition: Face images of actress Jennifer Grey demonstrating variations due to pose, illumination, expression, as well as alterations due to plastic surgery, aging and makeup. Images obtained from www.imdb.com, www.nydailynews.com and YouTube. The numbers represent the similarity match scores between (a) and each sample image (including sample (a)) as computed by a commercial face recognition system.
of heavy makeup (i.e., makeup invariant face recognition).

1.2 Challenges Due to Makeup

Makeup refers to the natural or synthetic chemical compounds used to change the appearance of a face. It is widely used and has become a daily necessity for many, as reported in a recent British poll of 2,000 women\(^6\), and as evidenced by a 3.6 Billion sales volume in 2011 in the United States\(^7\). Figure 1.8 illustrates the enormous market size of decorative cosmetics [3]. The cosmetic industry has introduced a number of skin, eye and lip makeup products (see Table 1.3). Skin makeup can be used to alter skin color, suppress wrinkles, and cover aging spots. Lip makeup is commonly used to shine the lips, alter its shape, and restore moisture to the lip. Eye makeup is widely used to change the shape of eyebrows, alter eye contour, and increase eye shadow [47]. We provide some examples in Figure 1.9.

Most people apply makeup to either cover facial defects such as wrinkles or pores, or to emphasize specific features. Others apply makeup to simply express personality or individuality [48]. The benefits of applying makeup have been elaborated in interviews conducted by Dellinger and Williams [49] including revitalized and healthy appearance, as well as increased credibility. The substantial effect of makeup application on an exemplar subject is visualized in Figure 1.10. Admittedly, makeup is a subjective matter and the understanding of makeup can vary from culture to culture [49]. Makeup is also reckoned as a soft biometric trait [50].

Makeup can be used to either emphasize or deemphasize facial features, thereby changing the appearance of a face image. The change is not only limited to color, but also to the texture and shape features of a face. Since face matching is typically done on gray-scale face images, changes due to texture and shape are more prominent than that of color. In this section, we introduce our work on the impact of makeup on both face recognition systems as well as soft biometric systems. While previous literature has referred to makeup as a confounding

\(^6\)http://www.superdrug.com/content/ebiz/superdrug/stry/cgq1300799243/survey_release - jp.pdf
Europe, United States, Japan and China
Makeup Product Market Size (€136.2 billon)

Figure 1.8: Makeup product market size in Europe, United States, Japan and China. The figure is generated based on the statistical data in report [3].

Table 1.3: Examples of face altering makeup items.

<table>
<thead>
<tr>
<th>Regions</th>
<th>Related makeup products</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skin Makeup</td>
<td>Cream, Blush, Bronzer, Concealer,</td>
</tr>
<tr>
<td></td>
<td>Foundation, Powder, Face Primer, Luminizer, Oil-Control Blotting</td>
</tr>
<tr>
<td>Eye Makeup</td>
<td>Brow Liner, Eyeliner, Eyeshadow, Eyebrows,</td>
</tr>
<tr>
<td></td>
<td>Eyelashes, Eye Brushes, Under Eye Concealer</td>
</tr>
<tr>
<td>Lip Makeup</td>
<td>Lipstick, Lip Gloss, Lip Plumper, Lip Stain</td>
</tr>
<tr>
<td></td>
<td>Lip Balm, Lip Color, Lip Brushes, Lip Liner</td>
</tr>
</tbody>
</table>

factor, majority of the system vendors have not addressed this problem, but have rather recommended users to avoid heterogeneous images\(^8\). This poses immediate concerns when subjects are not cooperative or not aware of the fact that makeup will affect the recognition process (Figure 1.10). Hence, it is of great interest to systematically study the impact of makeup on both face recognition and soft biometric classification systems\(^9\). Such an analysis will help in designing a robust makeup invariant face recognition system or a soft biometric classification system.

\(^8\)http://www.neurotechnology.com/face-image-recommendations-constraints.html

\(^9\)http://biometrics.org/bc2013/presentations/face_ross_thursday_1140.pdf
<table>
<thead>
<tr>
<th>Face Powder</th>
<th>Face Cream</th>
<th>Face Blush</th>
<th>Oil-Control Blotting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eyeliner</td>
<td>False Eyelashes</td>
<td>Eyebrow</td>
<td>Eye Brushes</td>
</tr>
<tr>
<td>Lip Stain</td>
<td>Lipstick</td>
<td>Lip Gloss</td>
<td>Lip Balm</td>
</tr>
</tbody>
</table>

Figure 1.9: Different makeup products; Images are from http://www.sephora.com.
1.2.1 Impact on Automated Face Recognition Algorithms

Makeup has received limited attention in the face recognition literature, although it has been alluded to superficially [5, 51, 52]. It is non-permanent and can alter the perceived shape and texture of a face, thereby compromising the accuracy of a face recognition system. Makeup is commonly used to improve the aesthetics of the face. However, it can strategically be used by an adversary for spoofing or obfuscation purposes (Fig. 1.11). In spoofing, an adversary modifies his facial appearance to look like another individual; in obfuscation, an adversary alters his appearance in order to evade recognition by a biometric system.

![Figure 1.10: An example of showing how makeup can be used to change the overall facial appearance, resulting in a possible false non-matching case. Image is obtained from the World Wide Web.](image)

![Figure 1.11: The effect of makeup. The subject in (a) is shown after the application of makeup in (b) - (f). Images are obtained from YouTube.](image)

Though makeup has been predominantly used by female subjects, it has generated in-
creasing interest in male clients (see Figure 1.12). Visually, the makeup effects are not as evident as those observed in female subjects. But it is not difficult to envision scenarios where makeup can change the overall appearance of male subjects as well.

Figure 1.12: Examples of a male subject before and after the application of makeup. From left to right, makeup varies from light to heavy. Images are obtained from YouTube.

Makeup is used by people across different cultures and race groups (see Figure 1.13). Different makeup products have been developed to cater to the needs of different groups. Caucasian, African and Asian women often need to search for products such as the right color, undertones, and pigments in order to match their skin color appropriately [53]. We later present a comprehensive study that investigates the impact of makeup on different face recognition algorithms and commercial systems. Further, the study presents a quantitative analysis of makeup on two different soft biometric traits: race and gender.

1.2.2 Impact on Gender Estimation Algorithm

Gender estimation is a fundamental task for human beings, as many social activities depend on the successful perception of gender information. In the realm of biometrics, gender is viewed as a soft biometric trait that can be used to index databases or enhance the recognition accuracy of primary traits such as face [54]. Gender estimation is considered as a binary classification problem: male or female. Often times, the way humans perceive gender does not only rely upon the perception of the face region, but also on the surrounding context, such as hair, dress and skin tone [54]. The scope of this study is only limited to the problem of estimating gender from cropped face images.

Given two face images that pertain to the same identity, it is expected that their gender will be the same, though there might be an age difference. However, the application of
makeup can change the perception of gender, resulting in false non-match of two images. Since gender estimation systems may be independently employed for access control, human computer interaction and demographical analysis [55], robust determination of gender under the influence of makeup is a matter of substantial interest to the field of biometrics. Very few research studies have considered the makeup factor in the prediction of human attributes [47]. In this work, we examine and measure the performance of gender estimation algorithm under the influence of makeup.

### 1.2.3 Impact on Age Estimation Algorithm

Age estimation is the task of automatically determining an exact age or age range from a given subject [56]. Potential applications of automatic age estimation include law enforcement, security control and human-computer interaction [57]. The face aging process is impacted not only by intrinsic factors (e.g., genetic factors), but also by extrinsic factors...
Figure 1.14: An example of showing how makeup can change the perception of gender, resulting in a possible gender estimation error. Images are obtained from YouTube.

(e.g., expression [58] and environment) [57]. However, we do not yet fully understand the relationship between age estimation and facial makeup changes. To develop a practical and robust age estimation system, it is imperative to investigate if an automated face-based age estimation algorithm is (a) influenced by the application of facial makeup, (b) how to quantify such an influence if it exists, and (c) if a robust solution could be developed to mitigate the problem caused by facial makeup.

1.3 Cross-spectral Matching Challenge

The focus of face recognition has been primarily centered on matching face images captured in the visible spectrum (between 0.4µm and 0.7µm wavelength). However, changes in lighting condition usually results in lower face matching performance [59], even in a cooperative environment. A face is a 3D object illuminated by ambient lighting source from different directions [60]. Hence, the appearance of a face can change dramatically when projected onto a 2D image space. The changes between two samples from the same identity under different illumination conditions can be larger than two samples acquired under two different identities [61].

The use of infrared (IR) spectrum band (0.7-14µm) provides a feasible solution for illumination invariant face recognition due to the fact that infrared spectrum is not sensitive to
changes in lighting in the visible spectrum. The infrared spectrum band can be separated into four main spectrum bands: near-infrared (NIR) (0.7-0.9\(\mu\)m), short-wave infrared (SWIR) (0.9-2.4\(\mu\)m), mid-wave infrared (MWIR) (3.0-5.0\(\mu\)m), and long-wave infrared (LWIR) (8.0-14.0\(\mu\)m) [60]. NIR and SWIR spectra are associated with reflected solar radiation, whereas MWIR and LWIR are associated with thermal radiation emitted from the objects. Therefore, NIR and SWIR are referred to as reflected IR, whereas MWIR and LWIR are referred to as thermal IR [60]. While there are many other spectral bands available (see Figure 1.15), spectral bands below VIS spectrum such as X-rays and ultraviolet are harmful to human and thus cannot be used for face recognition. Other spectral bands such as Microwave and Radio/TV waves are not applicable in face recognition. For notational convenience, we simply use thermal to denote thermal IR.

<table>
<thead>
<tr>
<th>Micrometer ((\mu)m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
</tr>
<tr>
<td>X-rays</td>
</tr>
<tr>
<td>10^{-4}</td>
</tr>
</tbody>
</table>

Figure 1.15: An illustration of different spectral bands. Visible, near-infrared and mid-wave infrared spectral bands are studied in this work.

In nighttime environments, near-infrared (NIR) and thermal (THM) image based face recognition technologies have been used to cope with different lighting conditions [61, 62]. The impact of ambient lighting is largely reduced by acquiring images in NIR and THM spectra. Thermal emissions from skin are an intrinsic property, which is considered to be independent of illumination. However, this requires that enrolled images be captured in the same spectral band as probe images. This becomes an imminent concern since images in legacy face databases are often captured in the visible (VIS) spectrum. Therefore, matching NIR or THM face images against VIS face images is of particular importance in designing nighttime face recognition systems [43]. This is referred to as cross-spectral face recogni-
Traditional face recognition often requires images to be captured and processed under similar conditions, e.g., visible vs visible (VIS vs VIS), near-infrared vs near-infrared (NIR vs NIR), and thermal vs thermal (THM vs THM). The matching performances start to drop significantly when the images to be compared are from different spectral bands (see Table 1.4 and Table 1.5). Additionally, cross-spectral matching at long stand-off distances further compromises the accuracy of such face recognition systems [7]. The objective of cross-spectral face recognition is to design robust algorithms that are capable of minimizing the impact of such spectral differences. We are interested in matching NIR against VIS and THM against VIS, considering the practical applications in nighttime surveillance and military operations [66]. Thermal images are often acquired from natural radiation emitted by the human face, in the mid-wave or long-wave infrared wavelength. On the other hand, for obtaining NIR images, active NIR lights are used to illuminate the face during image acquisition [67]. Visually speaking, the difference between NIR and VIS is less severe than the difference between THM and VIS. Despite significant progress made in face recognition systems, most existing commercial off-the-shelf (COTS) face recognition systems are not designed to deal with cross-spectral face recognition scenarios [65]. Hence, the need to develop face recognition algorithms that are specifically tailored to this task is of substantial interest. Two of the most representative methods for achieving this spectral invariance are the selection of feature descriptors that are stable between two different spectral bands, and learning features that can effectively compensate for such differences [43].

1.4 Thesis Contributions

The contributions of our research are to develop robust matching algorithms that can cope with heterogeneous face images. Towards addressing this goal we have conducted the following research activities.

- Analyzing the impact of facial cosmetics on face recognition. In this regard, we evaluate state-of-the-art face matchers and provide extensive experimental analysis to justify its impact.
Table 1.4: Illustration of face image samples captured across different spectra and distances. Images are from Long Distance Heterogeneous Face Database (LDHF-DB) [7].

<table>
<thead>
<tr>
<th>Distance</th>
<th>Subject 1 (VIS)</th>
<th>Subject 1 (NIR)</th>
<th>Subject 2 (VIS)</th>
<th>Subject 2 (NIR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1m</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>60m</td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
<tr>
<td>100m</td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td>150m</td>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
<td><img src="image15.png" alt="Image" /></td>
<td><img src="image16.png" alt="Image" /></td>
</tr>
</tbody>
</table>

- Analyzing impact of facial cosmetics on gender and age estimation. In this regard, we consider the use of facial cosmetics for (a) gender spoofing where male subjects attempt to look like females and vice versa, and (b) age alteration where female subjects attempt to look younger or older than they actually are.

- Developing a method to automatically detect the presence of makeup in face images, as well as present an adaptive pre-processing scheme that exploits knowledge of the presence or absence of facial makeup to improve the matching accuracy of a face matcher.

- Developing a patch-based ensemble learning method, which uses multiple subspaces generated by sampling patches from before-makeup and after-makeup face images, to address the problem of face recognition under the influence of makeup.

- Performing a comprehensive analysis on different types of image filters that can be used for cross-spectral face recognition and designing a scheme that is specifically used for remote heterogeneous face recognition.
Table 1.5: Illustration of face image samples captured under visible and thermal (MWIR: 3-5\(\mu\)m) spectra. Images are from Pinellas County Sheriff’s Office (PCSO) Thermal Face Dataset.

<table>
<thead>
<tr>
<th>Subject 1 (THM)</th>
<th>Subject 1 (VIS)</th>
<th>Subject 1 (VIS)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Subject 1 (THM) Image" /></td>
<td><img src="image2" alt="Subject 1 (VIS) Image" /></td>
<td><img src="image3" alt="Subject 1 (VIS) Image" /></td>
</tr>
<tr>
<td><img src="image4" alt="Subject 1 (THM) Image" /></td>
<td><img src="image5" alt="Subject 1 (VIS) Image" /></td>
<td><img src="image6" alt="Subject 1 (VIS) Image" /></td>
</tr>
<tr>
<td><img src="image7" alt="Subject 1 (THM) Image" /></td>
<td><img src="image8" alt="Subject 1 (VIS) Image" /></td>
<td><img src="image9" alt="Subject 1 (VIS) Image" /></td>
</tr>
</tbody>
</table>

- Developing a matching framework for face recognition between VIS and THM images based on a cascaded subspace learning scheme, which consists of whitening transformation, factor analysis, and common discriminant analysis.

1.5 Thesis Organization

In Chapter 2, we present experimental results that reveal the effect of makeup on automated face recognition and suggest that this simple alteration can indeed compromise
the accuracy of a biometric system. Our findings clearly indicate the need for a better understanding of this face altering scheme and the importance of designing algorithms that can successfully overcome the obstacle imposed by the application of facial makeup.

Chapter 3 further analyzes the impact of cosmetics on automated gender and age estimation algorithms. In this regard, we consider the use of facial cosmetics for (a) gender spoofing where male subjects attempt to look like females and vice versa, and (b) age alteration where subjects attempt to look younger or older than they actually are. Our findings suggest that facial cosmetics can confound automated gender and age estimation schemes.

Chapter 4 proposes a method to automatically detect the presence of makeup in face images. The proposed algorithm extracts a feature vector that captures the shape, texture and color characteristics of the input face, and employs a classifier to determine the presence or absence of makeup. Besides extracting features from the entire face, the algorithm also considers portions of the face pertaining to the left eye, right eye, and mouth. Experiments on two datasets consisting of 151 subjects (600 images) and 125 subjects (154 images), respectively, suggest that makeup detection rates of up to 93.5% (at a false positive rate of 1%) can be obtained using the proposed approach. Further, an adaptive pre-processing scheme that exploits knowledge of the presence or absence of facial makeup to improve the matching accuracy of a face matcher is presented.

In Chapter 5, we introduce a patch-based ensemble learning method, which uses multiple subspaces generated by sampling patches from before-makeup and after-makeup face images, to address the problem of makeup. In the proposed scheme, each face image is tessellated into patches and each patch is represented by a set of feature descriptors, viz., Local Gradient Gabor Pattern (LGGP), Histogram of Gabor Ordinal Ratio Measures (HGORM) and Densely Sampled Local Binary Pattern (DS-LBP). Then, a novel Semi Random Subspace Linear Discriminant Analysis (SRS-LDA) method is used to perform ensemble learning by sampling patches and constructing multiple common subspaces between before-makeup and after-makeup facial images. Finally, Collaborative-based and Sparse-based Representation Classifiers are used to compare feature vectors in this subspace and the resulting scores are combined via the sum-rule. The proposed face matching algorithm is evaluated on the YMU makeup dataset and is shown to achieve very good results. It outperforms other methods.
designed specifically for the makeup problem.

Finally in Chapter 6, we first propose a simple but effective approach for encoding Gradient information in Gabor-transformed images to represent the face, which can be used for identity, gender and ethnicity assessment. Extensive experiments on the standard face benchmark FERET (Visible versus Visible), as well as the heterogeneous face dataset HFB (Near-infrared versus Visible), suggest that the matching performance due to the proposed descriptor is comparable to state-of-the-art descriptor-based approaches in face recognition. Next, we introduce a multiresolution image representation scheme that is designed to deal with a specific problem, i.e., Remote Heterogeneous Face Recognition (RHFR). In the proposed scheme, three different image filtering methods are first used to reduce cross-spectral differences as well as noise contained in the degraded NIR images. Then, a patch-based image representation scheme that uses Pyramid-SIFT is used to describe face images captured under both VIS and NIR spectral bands. Finally, a Random Subspace Linear Discriminant Analysis (RS-LDA) method is utilized to learn multiple linear discriminative projections by performing random sampling on selected patches. The proposed face matcher is trained on the HFB Face Database and is evaluated on the Long Distance Heterogeneous Face Database (LDHF-DB). The reported results are promising, achieving significantly better results than existing methods. Finally, we introduce a Heterogeneous Face Recognition (HFR) matching framework, which uses multiple sets of subspaces generated by sampling patches from VIS and THM face images and subjecting them to a sequence of transformations, consisting of whitening transformation, factor analysis, and common discriminant analysis. The proposed face matching algorithm is evaluated on two multispectral face datasets and is shown to achieve very encouraging results.
Chapter 2

Impact of Cosmetics on Face Recognition

Motivated by the need to deploy highly reliable face recognition systems in security applications, we seek to initiate research that studies the impact of facial makeup on face recognition. Facial makeup is an example of a cosmetic alteration that can change the perceived appearance of the face. Other alterations include aging (a natural biological change) and plastic surgery (a medically induced change). Recent work has focused on the impact of plastic surgery on face recognition [68][69][70][71]. However, such surgical alterations are generally costly and permanent. On the other hand, non-permanent cosmetic alterations, such as makeup, tend to be simple, cost efficient and socially acceptable; at the same time, they have the potential to substantially change appearance. Specifically, such alterations can (a) alter the perceived facial shape by accentuating contouring techniques; (b) alter the perceived nose shape and size by contouring techniques; (c) enhance or reduce the perceived size of the mouth; (d) alter the appearance and contrast of the mouth by adding color; (e) alter the perceived form, color and location of eyebrows; (f) alter the perceived shape, size and contrast of the eyes; (g) conceal dark circles underneath the eyes; and (h) alter the perceived skin quality and color.

In addition to the aforementioned effects, cosmetics can also be used to successfully camouflage as well as affect wrinkles, birth moles, scars and tattoos. A vast cosmetics
market\(^1\) - typically targeted towards women - attempts to improve facial aesthetics while projecting good health. The beautification effects induced by cosmetics have been studied in recent research literature [72][73].

The impact of makeup on human ability to recognize faces has been studied by Ueda and Koyama [74]. The authors concluded that light makeup slightly increases human recognizability, whereas heavy makeup significantly decreases it. While the accentuation of distinctive characteristics using light makeup is helpful towards recognition, heavy makeup increases bilateral size and symmetry of the eyes and lips leading to a decreased characteristic distinctiveness of faces [75].

In spite of the aforementioned observations, there is no research that establishes the impact of cosmetic makeup on automated face recognition systems. While extensive research has been done to quantify the effect of pose, illumination and expression (commonly referred to as PIE) on face matching [59] [76], the biometric literature is largely silent about the effects of cosmetic makeup on face recognition. This is particularly of interest given that cosmetic makeup is commonly used by females in many parts of the world \(^1\). To address this aspect, we consider the following question: Can cosmetic facial makeup affect the matching accuracy of automated face recognition schemes? In order to answer this question, we first assemble two different face databases consisting of female subjects with and without makeup. Subsequently we test the matching accuracy of multiple face recognition algorithms (both academic and commercial) on these two databases. Experimental results suggest that face recognition can be substantially impacted by the application of facial makeup. To the best of our knowledge, this is the first work in the biometric literature to explicitly demonstrate this effect.

In this chapter, Section 2.1 discusses the databases that were assembled for this study. Section 2.2 describes the face recognition techniques that were used to assess the matching accuracy. Section 2.3 presents the experimental results. Section 2.4 discusses the key outcomes of this study. Section 2.5 addresses the challenge and provides a preliminary solution towards mitigating the effect of makeup on automated face recognition. Section 2.7 concludes the chapter.

\(^1\)http://www.japaninc.com/article.php?articleID=1390
2.1 Facial Makeup Databases

A review of the face recognition literature suggests that most databases available to biometric researchers are suitable for studying the effect of pose, illumination, expression, aging, plastic surgery, occlusion, motion blur, etc. on face recognition. However, there are no publicly available face databases that can be directly used to perform the study proposed in this work. The paper by Ueda and Koyama [74] discusses a proprietary database consisting of Japanese females. The work by Scherbaum et al. [48] specifies a private database containing 56 subjects, each with and without makeup. However, these databases could not be obtained from the respective researchers.

Hence, in order to conduct this study, we assembled two databases. The first database, which we refer to as the YouTube MakeUp (YMU) database, consists of face images of subjects gleaned from YouTube video makeup tutorials. The second database is a synthetic database, which we refer to as the Virtual MakeUp (VMU) database, where face images of Caucasian female subjects in the FRGC repository were synthetically modified to simulate the application of makeup. A publicly available software was used to perform this alteration. A brief description of these two databases is presented below.

2.1.1 YouTube Makeup (YMU) Database

We assembled a dataset consisting of 99 subjects, specifically Caucasian females, from YouTube makeup tutorials. We collected images of the subjects before and after the application of makeup. There are four shots per subject: two shots before the application of makeup and two shots after the application of makeup. For a few subjects, we were able to obtain three shots each before and after the application of makeup. We use the notation $N$ for the before makeup shots and $M$ for the after makeup shots.

The makeup in these face images varies from subtle to heavy as can be seen in Figure 2.1. The cosmetic alteration is mainly in the ocular area, where the eyes have been accentuated by diverse eye makeup products. Additional changes are on the quality of the

\[^2\text{http://www.nist.gov/itl/iad/ig/frgc.cfm}\]
\[^3\text{http://www.taaaz.com}\]
skin due to the application of foundation and change in lip color. This database includes some variations in expression and pose. The illumination condition is reasonably constant over multiple shots of the same subject. In few cases, the hair style before and after makeup changes drastically. URLs of videos from which face images were taken have been listed at http://www.antitza.com.

![Figure 2.1: Examples of two subjects in the YMU database assembled by the authors, each without [(a) and (c)] and with makeup [(b) and (d)]. The subject on the top has subtle makeup, while the one on the bottom has heavy makeup. The output of the automated face and eye detection scheme, see [4], is indicated in yellow.]

2.1.2 Virtual Makeup (VMU) Database

The VMU database was assembled by synthetically adding makeup to 51 female Caucasian subjects in the FRGC database. We added makeup by using a publicly available tool from Taaz. We created three virtual makeovers: (a) application of lipstick only; (b) application of eye makeup only; and (c) application of a full makeup consisting of lipstick, foundation, blush and eye makeup. Hence, the assembled dataset contains four images per subject: one before-makeup shot\(^4\) and three after-makeup shots. Figure 2.2 provides an example. We use the notation \(N\) for no-makeup, \(L\) for lipstick, \(E\) for eye makeup and \(F\) for

\(^{4}\)This image is referred to as the no-makeup shot
full makeup.

Figure 2.2: Example of a subject in the VMU database assembled by the authors. (a) depicts the subject without makeup, whereas, (b), (c) and (d) constitute the makeup shots. The makeup shots include the synthetic addition of (b) eye makeup (c) lipstick, and (d) full makeup, respectively, using the popular Taaz software.

The first database provides real life images of subjects with and without makeup, and helps assess the degradation in performance due to makeup. The second database contains images from the FRGC dataset that are synthetically processed by applying makeup on the portrayed subjects. The second database ensures that intra-class variations due to pose, illumination and expression are minimized. This allows us to focus solely on the analysis of the impact of makeup and also study of the impact of different kinds of makeup.

In the following section, we describe the face recognition methods that were used to evaluate the impact of facial makeup on face recognition systems.

### 2.2 Face Recognition Techniques

This section briefly introduces the three face recognition algorithms (i.e., matchers) that we employ in our study, viz., Gabor wavelets [77], Local Binary Pattern (LBP) [78] and the commercial Verilook Face Toolkit\(^5\). The choice of these matchers were based on the following observations:

- Gabor features encode both shape and texture information across different scales and orientations.

\(^5\)http://www.neurotechnology.com/verilook.html
- LBP captures micro-patterns and thus represents small-scale appearance.
- Verilook is a commercial face recognition system that has demonstrated competitive performance in several public face databases.

Descriptor-based face recognition methods, such as LBP, are computational and time efficient, and do not require a learning step for feature extraction and analysis. Thus, given a pair of face images, these methods can generate a matching score without requiring an explicit training phase. Hence, they were selected for the experiments conducted in this work.

Subspace based learning methods such as principal component analysis (PCA) and linear discriminant analysis (LDA) require the generation of a projection matrix. This projection matrix has to be computed from a set of training images. The result of the matching operation is, therefore, closely related to the choice of the training images used to generate the subspace. A different choice of training images can generate a different subspace, therefore biasing the matching results. To avoid this concern, we only consider descriptor-based schemes in our preliminary analysis.

Prior to invoking the three face matchers, individual images were converted from RGB to grayscale. Then, an automated face and eye detection routine was utilized to localize the spatial extent of the face in individual images. The face images were geometrically normalized based on the location of the eyes. Furthermore, we used Difference of Gaussian (DoG) to preprocess the images and minimize variations due to illumination. Therefore, all three face matchers operate on grayscale images of the face (see Figure 2.3).

### 2.2.1 Gabor Wavelets

Gabor wavelets are defined as follows [77]:

\[
\varphi_{\mu,\nu}(z) = \frac{|k_{\mu,\nu}|}{\sigma^2} e^{-\frac{|k_{\mu,\nu}||z||^2}{2\sigma^2}} \left[ e^{ik_{\mu,\nu}z} - e^{-\frac{\pi^2}{4}} \right],
\]

(2.1)

where \( \mu \) and \( \nu \) denote the orientation and scale of the Gabor kernels, \( z \) denotes the pixel position, i.e., \( z = (x, y) \), and \( \| \cdot \| \) denotes the norm operator [25]. The wave vector \( k_{\mu,\nu} \) is
given by:

\[ k_{\mu,\nu} = k_{\nu} e^{j\phi_{\mu}}, \]  

(2.2)

where \( k_{\nu} = k_{\text{max}} / f^\nu \) and \( \phi_{\mu} = \pi \mu / 8 \). Here, \( k_{\text{max}} \) is the maximum frequency and \( f \) is the spacing factor between kernels in the frequency domain. The subtraction of the component \( e^{-\sigma^2/2} \) imparts robustness to variations in illumination. The parameters \( \mu \) and \( \nu \) are chosen as 8 orientations and 5 scales, respectively, resulting in a total of 40 Gabor kernels. The size of Gabor kernels is determined by the parameter \( z \). The Gabor wavelet representation of an image is obtained by convolving it with these Gabor kernels:

\[ G_{u,v}(z) = I(z) * \varphi_{u,v}(z). \]  

(2.3)

The complex Gabor response has two parts, the real part \( \Re_{u,v}(z) \) and the imaginary part \( \Im_{u,v}(z) \). Such a convolution operation can be performed in the frequency domain via Fast Fourier Transform (FFT). The magnitude of the Gabor response (see an example in Figure 2.3 (c)) is computed as:

\[ A_{u,v}(z) = \sqrt{\Re_{u,v}(z)^2 + \Im_{u,v}(z)^2}. \]  

(2.4)

Figure 2.3: (a) Example makeup (M) image from the YMU database. (b) Preprocessed image. (c) Gabor magnitude response (40 images). (d) LBP coded image. We note that the Gabor response around the mouth and ocular regions is more prominent than in other parts of the face.

The Gabor features can be extracted by downsampling the output of each Gabor magnitude image, normalizing them to zero mean and unit variance, and concatenating them.
A simple $L_2$ norm distance measure ($\phi$) is used to compare the distance between Gabor features.

$$\phi(G^1, G^2) = \sqrt{\sum_{i=1}^{L} (g^1_i - g^2_i)^2},$$

(2.5)

where $G^1$ and $G^2$ are the Gabor features extracted from two images, and $L$ is the length of the feature vector.

### 2.2.2 Local Binary Pattern (LBP)

The LBP descriptor [23] was first proposed as a type of texture descriptor that characterizes micro-patterns or micro-structures in an image by binarizing a $3 \times 3$ neighborhood based on the differences in pixel intensity between the center pixel and neighborhood pixels, and converting the resulting binary string into a decimal value. In order to better represent large-scale structures in images, the small-scale patterns were extended to accommodate neighborhoods of different sizes. LBP has many desired properties such as tolerance to monotonic illumination changes and computational efficiency.

The binary pattern for pixels evenly lying in a neighborhood $(x_i, i = 0, 1, \ldots, P - 1)$ with respect to the center pixel $x_c$, is computed as follows:

$$f(x_i - x_c) = \begin{cases} 1 & \text{if } x_i - x_c \geq \tau; \\ 0 & \text{if } x_i - x_c < \tau. \end{cases}$$

(2.6)

Next, a binomial weight $2^i$ is assigned to each value of $f(x_i - x_c)$ in order to generate the LBP code:

$$LBP(x_c) = \sum_{i=0}^{P-1} f(x_i - x_c)2^i,$$

(2.7)

where $P$ is the total number of pixels in this neighborhood or region. As an example, for a $3 \times 3$ neighborhood, the value of $P$ will be 8. The threshold value $\tau$ is set to be zero for LBP. This results in an 8-bit code for each encoded pixel. The LBP operator is often denoted as $LBP_{P,R}$, where $P$ refers to the number of sampling points and $R$ is the radius of the neighboring region. To generate LBP features, each LBP coded image (see an example in Figure 2.3 (d)) is divided into sub regions, where histogram information is extracted from each sub region and then concatenated to form the descriptor. The histogram intersection
similarity measure (φ) that is used to compare two LBP descriptors is defined as follows [25],

\[
φ(H^1, H^2) = \sum_{i=1}^{L} \min(h^1_i, h^2_i),
\]

where \(H^1\) and \(H^2\) are the LBP features from images \(I^1\) and \(I^2\), respectively. \(L\) is the number of histogram bins. The similarity measure is then converted to a distance measure.

### 2.2.3 Neurotechnology Verilook

Verilook\(^6\) is a commercial face detection and recognition software, which computes matching scores between images, without a learning phase. While the underlying algorithm is not publicly disclosed, it is known that several face recognition algorithms have been combined in this software.

Having introduced the databases and the face matchers, we now proceed to describe the experimental component of the work.

### 2.3 Experiments

The purpose of the experiments was to determine the impact of facial makeup on the performance of automated face recognition algorithms. To do so, we compared the ability of such algorithms to recognize people before and after applying makeup. In this regard, we used the YMU database to understand the degradation in performance on real makeup images and the VMU database to determine which type of facial cosmetic alteration has the most impact on matching accuracy.

#### 2.3.1 Experiments on the YMU Database

The YMU database contains 99 subjects. There are four images associated with each subject: two images without makeup and two images with makeup. Let \(N_1\) and \(N_2\) denote the images without makeup, and \(M_1\) and \(M_2\) denote the images with makeup. Genuine and impostor scores for each of the three face matchers were generated according to the following protocol:

\(^6\)http://www.neurotechnology.com/verilook.html
1. Matching $N_1$ against $N_2$: Both the images to be compared do not have makeup (the before-makeup images).

2. Matching $M_1$ against $M_2$: Both the images to be compared have makeup (the after-makeup images).

3. Matching $N_1$ against $M_1$, $N_1$ against $M_2$, $N_2$ against $M_1$, $N_2$ against $M_2$: One of the images to be compared has no makeup while the other has makeup.

The EERs (Equal Error Rates) of the matching scenarios considered in the YMU database are summarized in Table 2.1. The general observation here is that EERs for the $N$ vs $M$ cases are substantially higher than for the $M$ vs $M$ and $N$ vs $N$ cases. The EER for the $N$ vs $N$ case is in the range of 6.50% (LBP) to 10.85% (Verilook), whereas for the $N$ vs $M$ case the EER moves up to 23.68% (Verilook). The results show that the application of makeup results in a substantial challenge for automated face recognition algorithms, which needs to be addressed in the interest of security. The performances of the academic face matchers outperform that of the commercial matcher; this might be due to the fact that Verilook does not offer a tuning option. Figure 2.4 shows a subject in the YMU database whose face images have each been normalized and cropped. The figure also reports the related distance scores generated by the three face matchers. These scores illustrate the drop in similarity (and an increase in distance score) when makeup is applied to a subject’s face.

Table 2.1: EER (%) of the face matchers on the YMU database. In all cases, comparing a makeup image ($M$) against a no-makeup image ($N$) decreases accuracy.

<table>
<thead>
<tr>
<th></th>
<th>$M$ vs $M$</th>
<th>$N$ vs $M$</th>
<th>$N$ vs $N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gabor</td>
<td>11.59%</td>
<td>21.47%</td>
<td>7.01%</td>
</tr>
<tr>
<td>LBP</td>
<td>9.41%</td>
<td>18.71%</td>
<td>6.50%</td>
</tr>
<tr>
<td>Verilook</td>
<td>13.55%</td>
<td>23.68%</td>
<td>10.85%</td>
</tr>
</tbody>
</table>

Figure 2.5 illustrates the boxplot of genuine score distributions for one of the face matchers LBP. Here, we note the shift in genuine scores when makeup is applied (the center boxplot). More specifically, the distance score increases when an image with makeup is com-
pared against an image without makeup. It is reasonable to state at this time that makeup has the potential to reduce the matching accuracy of face recognition algorithms thereby presenting an increased security risk.

![Sample subject from the YMU database.](image)

<table>
<thead>
<tr>
<th></th>
<th>(a) vs (b)</th>
<th>(a) vs (c)</th>
<th>(a) vs (d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N_1 vs N_2</td>
<td>0.404</td>
<td>0.573</td>
<td>0.527</td>
</tr>
<tr>
<td>N_1 vs M_1</td>
<td>0.447</td>
<td>0.505</td>
<td>0.487</td>
</tr>
<tr>
<td>N_1 vs M_2</td>
<td>0</td>
<td>0.399</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Figure 2.4: Sample subject from the YMU database. Intra-class variations can be observed between the no-makeup shots (a) and (b) as well as the makeup shots (c) and (d). In (e), the distance scores as reported by the three face matchers are presented for the various matching scenarios.

### 2.3.2 Experiments on the VMU Database

The VMU database contains 51 subjects. Each subject has one no makeup image and 3 after-make up images. Let \( N \) denote the image without makeup. Let \( L, E \) and \( F \) denote, respectively, images with lipstick only, with eye makeup only, and with full makeup. The following configurations were considered for generating genuine and impostor matching scores for each of the three face matchers.

1. Matching \( N \) against \( L \): An image without makeup is compared against the same image with lipstick added.

\(^7\)We note that there are intra-class variations for both the \( N \) versus \( N \) and \( M \) versus \( M \) cases, which cause a decrease in the performance of the face matchers for those cases as well.
2. Matching $N$ against $E$: An image without makeup is compared against the same image with eye makeup added.

3. Matching $N$ against $F$: An image without makeup is compared against the same image with full makeup added.

The EER values corresponding to the aforementioned scenarios are reported in Table 2.2. Note that the impact of eye makeup is more pronounced than that of lipstick on the performance of the three face matchers. Figure 2.6 shows a sample subject from the VMU database whose faces have been normalized and cropped. The figure also reports the distance scores generated by the three face matchers.

Table 2.2: EER (%) of the face matchers on the VMU database. Matching a no-makeup image ($N$) against a full makeup image ($F$) results in decreased performance.

<table>
<thead>
<tr>
<th></th>
<th>$N$ vs $E$</th>
<th>$N$ vs $L$</th>
<th>$N$ vs $F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gabor</td>
<td>8.56%</td>
<td>8.15%</td>
<td>11.38%</td>
</tr>
<tr>
<td>LBP</td>
<td>4.32%</td>
<td>3.43%</td>
<td>4.79%</td>
</tr>
<tr>
<td>Verilook</td>
<td>25%</td>
<td>4.79%</td>
<td>30%</td>
</tr>
</tbody>
</table>

Figure 2.5: Boxplot of the genuine score distributions for the YMU database as computed using the LBP face matcher for the 3 matching scenarios: $M_1$ vs $M_2$, $M$ vs $N$, $N_1$ vs $N_2$.

Figure 2.7 illustrates the boxplot of genuine score distributions for the LBP face matcher. Here, we note the shift in genuine scores between the different makeup styles (see the center
Figure 2.6: Example of normalized and cropped images for a subject in the VMU database. (a) No makeup. (b) Eye makeup. (c) Lipstick only. (d) Full makeup. In (e), the distance scores as reported by the three face matchers are presented for the various matching scenarios.

<table>
<thead>
<tr>
<th></th>
<th>(a) vs (b)</th>
<th>(a) vs (c)</th>
<th>(a) vs (d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gabor</td>
<td>0.518</td>
<td>0.453</td>
<td>0.543</td>
</tr>
<tr>
<td>LBP</td>
<td>0.468</td>
<td>0.449</td>
<td>0.469</td>
</tr>
<tr>
<td>Verilook</td>
<td>1</td>
<td>0</td>
<td>0.915</td>
</tr>
</tbody>
</table>

More specifically, we observe that the distance score due to lipstick is the lowest, while that for full makeup is the highest. We conclude that applying makeup to the ocular region (either separately or in the context of a full makeup) has a higher potential to reduce the matching accuracy of face recognition algorithms.

Figure 2.7: Boxplot of the genuine score distributions as computed by the LBP face matcher on the VMU database for the 3 matching scenarios: $N$ vs $E$, $N$ vs $L$, and $N$ vs $F$. 
2.4 Observations

In this section we summarize the observations made from the experimental results.

- Although alterations due to cosmetic facial makeup are predominantly color-based, they clearly affect the performance of face matchers based on grayscale images.

- While the YMU database contains examples of natural makeovers, the VMU database depicts subjects with synthetic makeovers. However, the accuracy of face matchers decreases in both these databases when a face image without makeup is compared against a face with makeup. This is a clear indication that facial makeup introduces changes in the face that can affect the accuracy of a face recognition system.

- The impact due to the application of eye makeup is indicated to be the most pronounced. Individuals attempting to obfuscate (i.e., camouflage) their identity from a face recognition system may be able to do so by incorporating non-permanent cosmetic alterations to their ocular region. Since makeup is a socially acceptable cosmetic modification in most societies, these results indicate the possibility of compromising the security of a face or periocular biometric system [79].

- In this work, the application of makeup was not intended to deliberately undermine the security of a biometric system. However, it is not difficult to envision situations where an individual might utilize facial cosmetics to deliberately either obscure their identity or to impersonate another individual.

2.5 Addressing the Problem

Here we investigate an existing algorithm known as Local Gabor Binary Pattern (LGBP) [25], which encodes a series Gabor filtered images (as opposed to the original image) using LBP. We anticipate this method to be resilient, to a certain degree, to the effects of makeup. The LGBP matcher has the following steps:

1. The normalized face image is first convolved with a set of Gabor filters to generate the Gabor filter responses (Gabor magnitudes only) \( G_1, G_2, \ldots, G_{40} \).
2. The LBP coding method is applied to encode the filter responses resulting in 40 LBP maps $L_1, L_2, \ldots, L_{40}$.

3. Each $L_i$ is tessellated into 64 sub-regions, $S_{i1}, S_{i2}, \ldots, S_{i64}$, where the size of each sub-region is $16 \times 16$.

4. For each sub-region $S_{ij}$, $i = \{1, 2, \ldots, 40\}$, $j = \{1, 2, \ldots, 64\}$, a histogram, $H_{ij}$ is extracted, which has 16 levels (bins). Each histogram value can be denoted as $H_{ijl}$, $l = \{1, 2, \ldots, 16\}$.

5. For two face images, $I_1$ and $I_2$, their respective histograms $H_1$ and $H_2$ are compared as $\phi(H_1, H_2) = \sum_{i=1}^{40} \sum_{j=1}^{60} \sum_{l=1}^{16} \min(H_{ijl}^1, H_{ijl}^2)$.

A histogram-type representation considers both the local and global level features and can exhibit robustness to certain variations in the image. Examples of LBP maps generated by this approach is shown in Figure 2.8.

Figure 2.8: LGBP matcher. (a) Preprocessed sample makeup image from the YMU database. (b) The 40 LBP maps generated after applying LBP to 40 different Gabor magnitude responses.

The LGBP face matcher is applied to both the YMU and VMU databases. See Tables 2.3 and 2.4. The results are summarized below:

1. In the YMU database, LGBP results in an EER of 15.89% for the N vs M matching case (no make up against makeup) compared to 18.71% by the LBP approach and 21.47% by the Gabor approach.
2. In the VMU database, LGBP results in EERs of 5.44% for $N \text{ vs } E$, 2.90% for $N \text{ vs } L$ and 5.42% for $N \text{ vs } F$. Compared to the LBP, Gabor and VeriLook matchers (see Table 2.2), the LGBP matcher results in better accuracy.

<table>
<thead>
<tr>
<th></th>
<th>$M \text{ vs } M$</th>
<th>$N \text{ vs } M$</th>
<th>$N \text{ vs } N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGBP</td>
<td>7.58%</td>
<td>15.89%</td>
<td>3.78%</td>
</tr>
</tbody>
</table>

Table 2.4: EER (%) of LGBP matcher on the VMU database.

<table>
<thead>
<tr>
<th></th>
<th>$N \text{ vs } E$</th>
<th>$N \text{ vs } L$</th>
<th>$N \text{ vs } F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGBP</td>
<td>5.44%</td>
<td>2.90%</td>
<td>5.42%</td>
</tr>
</tbody>
</table>

The ROC curves of all four algorithms on the YMU database are presented in Figure 2.9. We can clearly see the performance degradation of the Gabor, LBP and Verilook matchers in the $N \text{ vs } M$ case. LGBP outperforms the other algorithms, with an EER of 15.89% the presence of makeup remains though a challenge for face recognition algorithms.

We have conducted experiments on an extended version of YMU dataset (151 subjects, 604 images) with seven commercial and academic algorithms: COTS-1, COTS-2, COTS-3, OpenBR [80], Local Gabor Binary Pattern (LGBP) [25], Local Gradient Gabor Pattern (LGGP) [81] and Histogram of Monogenic Binary Pattern (HMBP) [82]. COTS-1, COTS-2 and COTS-3 are three commercial algorithms. Table 5.2 summarizes EERs (Equal Error Rates) of seven face matchers on this extended YMU dataset.

2.6 Influence of Demographics

The impact of cosmetics on face recognition has been thoroughly studied. However, the influence of demographics has not been investigated when cosmetics are analyzed. Here, we assemble two ethnicity datasets (African American and Asian) and one gender dataset (Male) in this study. The same experimental protocol in YMU experiment is adopted. Figure 2.10 and Figure 2.11 demonstrate the impact of cosmetics on face recognition across
Figure 2.9: Performance of the four face matchers on the YMU database. (a) Gabor; (b) LBP; (c) Verilook; (d) LGBP. We observe consistent performance decrease in the case $M$ vs $N$ for all 4 algorithms. LGBP provides the best results with an EER of 15.89%, this error rate suggests that makeup remains a challenge and requires more research.
Table 2.5: Equal Error Rates (%) corresponding to the seven face matchers on an extended version of YMU dataset with 151 subjects and 604 images.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>N vs N</th>
<th>M vs M</th>
<th>N vs M</th>
</tr>
</thead>
<tbody>
<tr>
<td>COTS-1</td>
<td>3.84</td>
<td>7.07</td>
<td>12.04</td>
</tr>
<tr>
<td>COTS-2</td>
<td>0.69</td>
<td>1.32</td>
<td>7.69</td>
</tr>
<tr>
<td>COTS-3</td>
<td>0.11</td>
<td>3.29</td>
<td>9.17</td>
</tr>
<tr>
<td>OpenBR</td>
<td>6.87</td>
<td>16.44</td>
<td>25.20</td>
</tr>
<tr>
<td>LGBP</td>
<td>5.34</td>
<td>8.77</td>
<td>19.71</td>
</tr>
<tr>
<td>LGGP</td>
<td>5.35</td>
<td>8.01</td>
<td>19.70</td>
</tr>
<tr>
<td>HMBP</td>
<td>6.25</td>
<td>10.87</td>
<td>21.54</td>
</tr>
</tbody>
</table>

different ethnicity groups. We do not observe much difference in EER when the impact of cosmetics on face recognition is compared against Caucasian. On the other hand, Figure 2.12 shows the impact of cosmetics on face recognition with gender demographics (male) considered. Compared to the results reported in previous section with female subset, we observe a significant decrease in EER. This is mainly due to less widespread use of facial cosmetics by male subjects. This suggests that makeup impact is consistently observed across different ethnicity and gender groups. And such an impact is mainly due to the amount of makeup that are applied.

2.7 Summary

In this work we presented preliminary results on the impact of facial makeup on automated face recognition. This is the first work that explicitly establishes the impact of facial makeup on automated biometric systems. We illustrated that non-permanent facial cosmetics can significantly change facial appearance, both locally and globally, by altering color, contrast and texture. Existing face matchers, which rely on such contrast and texture information for establishing a match, can be impacted by the application of facial makeup. We provided clear experimental evidence of the decrease in matching accuracy when makeup is used. Further, we designed a method to mitigate the effect of makeup on matching per-
Figure 2.10: Performance of the three face matchers on the Asian dataset. (a) LGBP (16.30% EER); (b) LGGP (16.38% EER); (c) HMBP (15.84% EER). We observe consistent performance decrease in the case $\mathcal{M}$ vs $\mathcal{N}$ for all 3 algorithms.
Figure 2.11: Performance of the three face matchers on the African dataset. (a) LGBP (17.54% EER); (b) LGGP (16.08% EER); (c) HMBP (17.89% EER). We observe consistent performance decrease in the case $M$ vs $N$ for all 3 algorithms.
Figure 2.12: Performance of the three face matchers on the Male dataset. (a) LGBP (7.63% EER); (b) LGGP (8.06% EER); (c) HMBP (6.69% EER). We observe consistent performance decrease in the case $\mathcal{M}$ vs $\mathcal{N}$ for all 3 algorithms.
formance. Future work will involve establishing a more detailed experimental protocol that quantifies the *degree* of makeup applied to a subject’s face.
Chapter 3

Impact of Cosmetics on Gender and Age

Recent studies have demonstrated the negative impact of facial cosmetics on the matching accuracy of automated face recognition systems [5, 51]. Such an impact has been attributed to the ability of makeup to alter the perceived shape, color and size of facial features, and skin appearance in a simple and cost efficient manner [5].

The impact of makeup on human perception of faces has received considerable attention in the psychology literature. Specifically, the issues of identity obfuscation [74], sexual dimorphism [83], and age perception [84] have been analyzed in this context. Amongst other things, these studies show that makeup can lead to higher facial contrast thereby enhancing female-specific traits [83], as well as smoothen and even out the appearance of skin thereby imparting an age defying effect [85]. This leads us to ask the following question: can makeup also confound computer vision algorithms designed for gender and age estimation from face images? Such a question is warranted for several reasons. Firstly, makeup is widely used and has become a daily necessity for many, as reported in a recent British poll of 2,000 women\(^1\), and as evidenced by a 3.6 Billion sales volume in 2011 in the United States\(^2\). Secondly,
a number of commercial software have been developed for age and gender estimation\(^3,4,5\). Thus, it is essential to understand the limitations of these software in the presence of facial makeup. Thirdly, due to the use of such software in surveillance applications [86], anonymous customized advertisement systems\(^6\) and image retrieval systems [87], it is imperative that they account for the presence of makeup if indeed they are vulnerable to it. Fourthly, gender and age have been proposed as soft biometric traits in automated biometric systems [88]. Given the widespread use of facial cosmetics, understanding the impact of makeup on these traits would help in accounting for them in biometric systems. Hence, the motivation of this work is to quantify the impact of makeup on gender and age estimation algorithms.

However, there is little work establishing the impact of cosmetics on gender and age estimation algorithms. Only one recent publication has considered the effect of makeup on age estimation [47], where an age index was used to adjust parameters in order to improve the system’s accuracy.

In this work, we seek to answer the following questions:

- Can facial makeup be used to spoof gender with respect to an automated gender estimation algorithm?

- Can the use of facial makeup confound an automated age estimation algorithm?

Towards answering these questions, we first assemble two datasets consisting of a) male subjects applying makeup to look like females and vice-versa, and b) female subjects applying makeup to conceal aging effects. Subsequently, we test gender and age estimation algorithms on these two datasets, respectively. Experimental results suggest that gender and age estimation systems can be impacted by the application of facial makeup. To the best of our knowledge, this is the first work to systematically demonstrate these effects. The results appear intuitive, since humans may have similar difficulties in estimating gender and age after the application of makeup. However, as reported in a recent study in the context of

\(^3\)www.neurotechnology.com/face-biometrics.html
\(^4\)www.visidon.fi/en/Face_Recognition#3
\(^5\)www.cognitec-systems.de/FaceVACS-VideoScan.20.0.html
\(^6\)articles.latimes.com/2011/aug/21/business/la-fi-facial-recognition-20110821
face recognition [89], human perception and machine estimation can be significantly different. This becomes especially apparent when only cropped images of the face are considered, without the surrounding hair and body information. In this work, only cropped face images are used for assessing impact of makeup on automated gender and age estimation algorithms.

The rest of the chapter is organized as follows. Section 3.1 introduces the problem of makeup-based gender alteration, presents the assembled dataset in Section 3.1.1, discusses the employed estimation algorithms in Section 3.1.3, and reports related results in Section 3.1.4. Section 3.2 introduces the problem of makeup induced age alteration, presents the assembled dataset in Section 3.2.1, discusses the employed age estimation algorithm in Section 3.2.3, and summarizes the results in Section 3.2.4. Section 3.3 discusses the results and Section 3.4 concludes the chapter.

### 3.1 Makeup Induced Gender Alteration

Interviews conducted by Dellinger and Williams [49] suggested that women used makeup for several perceived benefits including revitalized and healthy appearance, as well as increased credibility. However, makeup can also be used to alter the perceived gender, where a male subject uses it to look like a female (Figure 3.1(a) and Figure 3.1(b)), or a female subject uses it to look like a male (Figure 3.1(c) and Figure 3.1(d)). The makeup in both cases is used to conceal original gender specific cues and enhance opposite gender characteristics. For instance, in the male-to-female alteration case, the facial skin is first fully covered by foundation (to conceal facial hair and skin blemishes), and then eye and lip makeup (e.g., eye shadow, eye kohl, mascara and lipstick) are applied in the way females usually do. In the female-to-male alteration case, the contrast in the eye and lip areas is decreased using foundation, skin blemishes and contours (e.g., around the nose) are added (e.g., by using brown eye shadow), and male features such as mustache and beard are simulated (e.g., by using eye kohl).

We study the potential of such cosmetic applications to confound automatic face-based gender classification algorithms that typically rely on the texture and structure of the face image to distinguish between males and females [90]. While some algorithms [91] might
also exploit cues from clothing, hair, and other body parts for gender prediction, in this study we consider only the facial region. Therefore, we focus only on the cropped face, which minimizes the inclusion of factors such as hair, clothing and other accessories (see Figure 3.3).

Figure 3.1: Examples of subjects applying facial makeup for gender spoofing (from YouTube). Male-to-female (a-b): foundation conceals facial hair and skin blemishes; eye and lip makeup are then applied in the way females usually do. Female-to-male (c-d): dark eye-shadow is used to contour the face shape and the nose; then, thicker eye-brows, mustache, and beard are simulated using special makeup products. Only the facial region is used in this study (see Figure 3.3).

3.1.1 Makeup Induced Gender Alteration (MIGA) Dataset

To study makeup induced gender alteration, we searched the Web and assembled a dataset consisting of two subsets:

- Male subset consisting of 120 images of 30 subjects (2 before makeup and 2 after makeup images per subject): male subjects apply makeup to look like females,
• Female subset consisting of 128 images of 32 subjects (2 before makeup and 2 after makeup images per subject): female subjects apply makeup to look like males.

Figure 3.2: Example images from the Makeup Induced Gender Alteration (MIGA) dataset: (a) male-to-female subset: male subjects apply makeup to look like females, and (b) female-to-male subset: female subjects apply makeup to look like males. In both (a) and (b), the images in the upper row are before makeup and the ones below are the corresponding images after makeup.

The images were obtained from makeup transformation tutorials posted on YouTube, and the images exhibit differences in illumination and resolution, while subjects exhibit differences in race, facial pose and expression (see Figure 3.2). Note that the subjects were not trying to deliberately mislead automated systems. Despite the relatively small size of the dataset, it enables us to investigate the potential of makeup to confound computer vision-based gender classification systems.
3.1.2 Gender Classification and Alteration Metrics

For performance evaluation of gender classification systems, we define two classification rates:

- **Male Classification Rate**: the percentage of images (before or after makeup) that are classified as male by the gender classifier.

- **Female Classification Rate**: the percentage of images (before or after makeup) that are classified as female by the gender classifier.

Additionally, we introduce a metric called *gender spoofing index (GSI)* that quantifies the success of cosmetic induced gender spoofing. Let \( \{S_1, \cdots, S_n\} \) be a set of face images,
and let the corresponding label values be \( \{ \nu_1, \cdots, \nu_n \} \), where \( \nu_i \in \{ 0, 1 \} \), with 0 indicating male and 1 indicating female. Let \( \{ M_1, \ldots, M_n \} \) denote the images after the application of makeup. If \( G \) denotes the gender classification algorithm, then \( GSI \) is defined as:

\[
GSI = \sum_{i=1}^{\ell} I(G(S_i) \neq G(M_i)),
\]

where \( G(S_i) \) and \( G(M_i) \) are the gender labels as computed by the algorithm for \( S_i \) and \( M_i \), respectively, \( \ell = \sum_{i=1}^{n} I(G(S_i) = \nu_i) \) denotes the number of face images before makeup that were correctly classified by the algorithm and \( I(x) \) is the indicator function, where \( I(x) = 1 \) if \( x \) is true and 0 otherwise. In summary, \( GSI \) represents the percentage of face images whose gender prediction labels were changed after the application of makeup for those face images whose before makeup labels were correctly predicted.

Our hypothesis is that, if makeup can be used for gender spoofing, then the male classification rate will decrease after male-to-female alteration; and the female classification rate will decrease after female-to-male alteration.

### 3.1.3 Gender Estimation Algorithms

To study the effectiveness of makeup induced gender spoofing, we annotate the eyes of the subjects, crop the images to highlight the face region only (see Figure 3.3) and utilize three state-of-the-art gender classification algorithms (academic and commercial).

**Commercial Off-the-Shelf (COTS):** COTS is a commercial face detection and recognition software, which includes a gender classification routine. While the underlying algorithm and the training dataset that were used are not publicly disclosed, it is known that COTS performs well in the task of gender classification. To validate this, we first perform an experiment on a face dataset\(^7\) consisting of 59 male and 47 female faces that is a subset of the FERET database and which has been used extensively in the literature for evaluating gender classifiers. COTS obtains male and female classification accuracies of 96.61% and 97.87%, respectively, on this dataset. The system does not provide a mechanism to re-train the algorithm based on an external dataset; instead it is a black box that outputs a label.

\(^7\)www.cs.uta.fi/hci/mmig/vision/datasets/
(i.e., male or female) along with a confidence value.

**Adaboost:** The principle of Adaboost [87] is to combine multiple weak classifiers to form a single strong classifier as 

$$ y(x) = \sum_{t=1}^{T} \alpha_t h_t(x), $$

where \( h_t(x) \) refers to the weak classifiers operating on the input feature vector \( x \), \( T \) is the number of weak classifiers, \( \alpha_t \) is the corresponding weight for each weak classifier and \( y(x) \) is the classification output. In this work, feature vector \( x \) consists of pixel values from a \( 24 \times 24 \) image of the face. For every pair of feature values \((x_i, x_j)\) in the feature vector \( x \), five types of weak binary classifiers are defined:

$$ h_t(x) \equiv \{ g_k(x_i, x_j) \}, $$

where \( i, j = 1 \ldots 24, \quad i \neq j, \quad k = 1 \ldots 5 \), and

$$ g_k(x_i, x_j) = 1, \quad \text{if } (x_i - x_j) > t_k, $$

where \( t_1 = 0, \ t_2 = 5, \ t_3 = 10, \ t_4 = 25 \) and \( t_5 = 50 \). By changing the inequality sign in (3) from \( > \) to \( < \), another five types of weak classifiers can be generated, resulting in a total of 10 types of weak classifiers. Since \( g_k \) is non-commutative, \( g_k(x_i, x_j) \neq g_k(x_j, x_i) \), the total number of weak classifiers for a pair of features \( x_i \) and \( x_j \) is 20. The total number of weak classifiers selected by the AdaBoost algorithm, \( T \), is 1,000.

In order to utilize the Adaboost method for gender classification, the AR database\(^8\) (350 males and 350 females) was used to train the gender classifier. Each image is rescaled to \( 24 \times 24 \) and the column vectors consisting of pixel values are concatenated together to form the feature vector \( x \). Adaboost obtains male and female classification accuracies of 86.44\% and 82.98\%, respectively, on the FERET subset\(^7\).

**OpenBR:** OpenBR [80] is a publicly available toolkit for biometric recognition and evaluation. The gender classification algorithm utilized in this toolkit is based on [92]. An input face image is represented by extracting histograms of local binary pattern (LBP) and scale-invariant feature transform (SIFT) features computed on a dense grid of patches. The histograms from each patch are then projected onto a subspace generated using PCA in order

---

\(^8\)www2.ece.ohio-state.edu/\%7Ealeix/ARdatabase.html
to obtain a feature vector. Support Vector Machine (SVM) is used for classification. The
efficacy of the OpenBR gender classification algorithm is again validated on the FERET
subset indicated above. It attains accuracies of 96.91% and 82.98% for male and female
classification, respectively. The overall true classification rate is 90.57%, which outperforms
the other algorithms (Neural Network, Support Vector Machine, etc.) reported in [93].

3.1.4 Gender Spoofing Experiments

We conduct two experiments, corresponding to the two subsets in MIGA: (a) male-
to-female spoofing; and (b) female-to-male spoofing. For experiment (a) we report the
male classification rates, and for experiment (b) we report the female classification rates,
as elaborated in Section 3.1.2. In either case, the accuracy of gender classification before
and after the application of makeup are independently determined. Figure 3.4 presents the
output of COTS on some sample images before and after makeup.

a) Male-to-female alteration: We report the classification rates of the three gender
classification algorithms on the first subset of the MIGA dataset in Table 3.1. COTS obtains
a 68.33% male classification rate before makeup and 6.67% after makeup. The GSI value is
95.12%. This suggests that for most of the correctly classified male subjects before makeup,
the application of makeup alters the gender from the perspective of the commercial system.
We observe that AdaBoost has a male classification rate of 78.33% on the before makeup
images and 30% on the after makeup images. The GSI value for AdaBoost was 70.21%,
which suggests that makeup was successful in altering the perceived gender of a good number
of face images. OpenBR results in a similar trend where the male classification rate drops
from 55.0% to 15.0% after makeup. The corresponding GSI value is 75.76%.

b) Female-to-male alteration: We report the classification rates of the three gender
classification algorithms on the second subset of the MIGA dataset in Table 3.2. For Ad-
aboost, the female classification rate decreases from 53.13% before makeup to 9.37% after
makeup. A GSI of 88.24% indicates that gender alteration, with respect to the classifier,
was successful for a good number of subjects whose before makeup images was correctly
classified as female. COTS has a 100% female classification rate before makeup, which drops
Table 3.1: Male classification rates (%) and $GSI$ values (%) corresponding to the three gender classification algorithms on the male-to-female subset of the MIGA dataset.

<table>
<thead>
<tr>
<th></th>
<th>Before Makeup</th>
<th>After Makeup</th>
<th>$GSI$</th>
</tr>
</thead>
<tbody>
<tr>
<td>COTS</td>
<td>68.33</td>
<td>6.67</td>
<td>95.12</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>78.33</td>
<td>30.0</td>
<td>70.21</td>
</tr>
<tr>
<td>OpenBR</td>
<td>55.0</td>
<td>15.0</td>
<td>75.76</td>
</tr>
</tbody>
</table>

Figure 3.4: The output of the COTS gender classifier on the images shown in Figure 3.2. M indicates “classified as male” and F indicates “classified as female”. While all male-to-female transformations were successful in this example, only the leftmost and rightmost female-to-male transformations were successful.
Table 3.2: Female classification rates (%) and GSI values (%) corresponding to the three gender recognition algorithms on the female-to-male subset of the MIGA dataset.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Before Makeup (%)</th>
<th>After Makeup (%)</th>
<th>GSI (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>53.13</td>
<td>9.37</td>
<td>88.24</td>
</tr>
<tr>
<td>COTS</td>
<td>100.0</td>
<td>39.06</td>
<td>60.94</td>
</tr>
<tr>
<td>OpenBR</td>
<td>71.88</td>
<td>46.87</td>
<td>47.83</td>
</tr>
</tbody>
</table>

to 39.06% after the application of makeup. OpenBR obtains a 71.88% female classification rate before makeup, which drops to 46.87% after makeup. The GSI values for COTS and OpenBR are 60.94% and 47.83%, respectively.

We note from these experiments that some subjects can successfully apply makeup to alter the perceived gender, thereby misleading gender classification systems. Interestingly, we observe that female-to-male alteration is slightly more challenging than its counterpart. The difference can be noticed in the GSI values (see Table 3.1 and Table 3.2); specifically the male-to-female subset has a higher GSI value (e.g., COTS and OpenBR). A possible explanation for this observation is that male characteristics are easier to be concealed using makeup (e.g., heavy makeup), than to be created. However, it must also be noted that the three algorithms perform very differently in the gender classification task. The differential performance observed across the three algorithms on male/female classification rates could be due to the implicit differences in the features that they employ and the dataset used to train the individual algorithms.

### 3.2 Makeup Induced Age Alteration

Makeup can also be used to alter the perceived age of a person. This is accomplished by concealing wrinkles and age spots (using light foundation and a concealer), by brightening wrinkle-induced shadows in eye, nose and mouth regions (using concealer and powder), and by highlighting and coloring cheeks (using highlighter and blush). One reason for women to wear makeup is to increase their perceived competence and credibility [94]. In other cases, the goal of applying makeup is to make a subject look younger, while in the case of young
subjects the opposite effect might be desired [94]. Here, we seek to observe the impact of makeup on an automated age estimation algorithm. We minimize other confounding factors (e.g., hair and accessories) by using cropped and aligned faces.

3.2.1 Description of Datasets

We first conduct experiments on the YMU and VMU datasets [5], which were originally used to study the impact of makeup on automatic face recognition algorithms\(^9\). YMU consists of face images of 151 young Caucasian females obtained from YouTube makeup tutorials. For each subject, there are two images before and two images after makeup. VMU contains face images of 51 female Caucasian subjects from the FRGC database, to which makeup was synthetically applied using the Taaz software [5]. In VMU three types of makeup were virtually applied: lipstick, eye, and full makeup; hence there are four images per subject (one before makeup, one with lipstick only, one with eye makeup only and one with full makeup). In YMU and VMU the application of makeup was primarily to improve facial aesthetics, i.e., they were not applied with the intention of defeating a biometric system.

Additionally, to study makeup induced age alteration, we assembled another dataset (MIAA - Makeup Induced Age Alteration) consisting of images downloaded from the World Wide Web. These images correspond to 53 subjects, with one image before and one after the application of makeup per subject (see Figure 3.5). While the ground truth of a subject’s age is not available, we estimate that the subjects are older than 30 years and that makeup is applied with the goal of both improving aesthetics as well as making subjects look younger. However, for our study, knowledge about the absolute age of the subject is not required, since we are only interested in age difference between the before and after makeup images as assessed by the algorithm. Since the before and after makeup images correspond to the same session, the primary confounding covariate between them is the presence or absence of makeup.

In summary, the role of makeup in YMU, VMU and MIAA datasets is different. Subjects in MIAA knowingly apply makeup to appear younger, while in YMU and VMU makeup is

---

\(^9\)www.antitza.com/makeup-datasets.html
not specifically used for anti-aging purpose.

Figure 3.5: Example image pairs from the MIAA dataset. Top row: images before makeup; Bottom row: corresponding images after makeup.

3.2.2 Age Estimation and Alteration Metrics

The performance of the automated age estimation algorithm is calculated by the *Mean Absolute Error (MAE)*: \( MAE = \frac{1}{N} \sum_{k=1}^{N} |g_k - \hat{g}_k|/N \). Here \( g_k \) is the ground-truth-age, \( \hat{g}_k \) the estimated age for the \( k \)-th image, and \( N \) the number of test images. Age alteration is measured by *Mean Absolute Difference (MAD)*, which is computed as \( MAD = \frac{1}{N} \sum_{k=1}^{N} |\hat{g}_b^k - \hat{g}_a^k|/N \). Here \( \hat{g}_b^k \) is the estimated age of the before-makeup image and \( \hat{g}_a^k \) is the estimated age of the after-makeup image.

Our hypothesis here is that the estimated image after makeup will be significantly different than the estimated image before makeup. If so, this would indicate that makeup has the ability to impact age estimation algorithms.

3.2.3 Age Estimation Algorithm

The age estimation software used in this work utilizes the same feature set as the gender classifier in OpenBR (see Section 3.1.3), along with SVM regression. We first evaluate the reliability of this algorithm on a large-scale dataset, namely a specific subset of Morph [95]. This dataset contains 10,000 images of subjects in the age range 20 to 75. There are four age groups: 20-40 (2,514 images), 40-50 (5,524 images), 50-60 (1,790 images) and 60-70 (172
images). The MAEs of the OpenBR algorithm on these groups are 5.46, 5.75, 6.47, and 7.88, respectively. We note that the algorithm performs better on the youngest age group (20-40). The algorithm obtains an MAE of 5.84 years on the entire test set (10,000 images). The best reported performance on the entire MORPH database is an MAE of 4.18 years as reported in [96] based on the kernel-based partial least squares regression method.

3.2.4 Age Alteration Experiments

First, we conduct age estimation experiments on YMU and VMU in order to a) show the impact of makeup on age estimation (YMU), and b) study this impact based on the type of makeup used (VMU). Towards this, we use the OpenBR software to estimate the age for all images and compute the differences in estimated age, before and after makeup, for each subject:

- Age difference of B versus B (B vs B): Both face images are before makeup.
- Age difference of A versus A (A vs A): Both face images are after makeup.
- Age difference of A versus B (A vs B): One of the face images is after makeup while the other is before makeup.

We observe that age differences between after makeup images and before makeup images (A vs B) are larger than in the other two cases (see Figure 3.6(a)). This suggests that makeup does have an impact on the performance of automated age estimation. Next, we perform age estimation on the VMU dataset and compute age differences corresponding to (N vs L): a face without makeup versus the same face with lipstick; (N vs E): a face without makeup versus the same face with eye makeup; and (N vs F): a face without makeup versus the same face with full makeup (foundation, eye and lip makeup). We observe that the use of lipstick (N vs L) has a lower impact on age estimation, compared to the application of eye makeup (N vs E) and full makeup (N vs F) (see Figure 3.6(b)).

Next, we conduct experiments on the MIAA dataset, in order to evaluate the age-defying effect of makeup on computer vision-based age estimation algorithms. To make this assessment, we use the difference rather than the absolute difference when comparing the
before and after makeup images. Results indicate that 56.61% of the subjects are estimated to be younger after the application of makeup. Specifically, 32.08% are estimated as being 5 or more years younger, with a maximum being 20 years younger (see Figure 3.7). In order to validate the significance of the above result, we perform a one-sided hypothesis test with $H_1 : \mu_b - \mu_a < 0$, where $\mu_b$ is the mean age for before makeup images and $\mu_a$ is the mean age for after makeup images. The null hypothesis is $H_0 : \mu_b - \mu_a = 0$. The one-sided hypothesis test rejects the null hypothesis at the 5% significance level. It is therefore evident that makeup does have an age-defying effect. Moreover, a MAD of 7.67 is obtained on the MIAA dataset, which is larger than the MAE value (5.84) obtained on the Morph dataset, thus suggesting that the change observed in estimated age is significant even after taking the error tolerance into account. The output of OpenBR on some sample images is presented in Figure 3.8.

### 3.3 Discussion

We summarize the main observations from the experiments conducted in this work:\footnote{Details about obtaining the MIGA and MIAA datasets will be posted at www.antitza.com/makeup-datasets.html}:

- 

Figure 3.7: Age defying effect of makeup on the OpenBR algorithm. x-axis values indicate the difference in estimated age in years (after makeup - before makeup), while y-axis values show the percentage of subjects in MIAA.

- Makeup induced gender spoofing does impact automated gender classification systems. The observation holds for male-to-female, as well as for female-to-male alterations. The female-to-male alteration was observed to be slightly more challenging than its counterpart.

- The application of makeup can impact automatic age estimation algorithms.

These observations point out the need for developing robust gender and age estimation methods that are less impacted by the application of makeup. There are several ways to potentially address this issue:

- Whenever makeup is detected, an image pre-processing scheme can be used to mitigate its effect, as was shown in the context of face recognition [97].

- As demonstrated in the work of Feng and Prabhakaran [47], the estimated age can be adjusted accordingly after makeup is detected.
The accuracy of gender and age estimation algorithms depends on the features used to represent the face, as well as the classifier used to estimate these attributes. Therefore, exploring different types of feature sets and classifiers will be necessary to devise a robust solution.

The training set used by the learning algorithms can be suitably populated with face images having makeup. This would help the algorithm learn to estimate gender and age in the presence of makeup.

Age (or gender) can be independently estimated using different components of the face and the independent estimates can be combined to generate a final estimate.

3.4 Summary

In this chapter we presented preliminary results on the impact of facial makeup on automated gender and age estimation algorithms. Since automated gender and age estimation schemes are being used in several commercial applications, this research suggests that the issue of makeup has to be accounted for. While a subject may not use makeup to intentionally defeat the system, it is not difficult to envision scenarios where a malicious user may employ...
commonly-used makeup to deceive the system. Future work will involve developing gender and age estimation algorithms that are robust to changes introduced by facial makeup.
Chapter 4

Automatic Facial Makeup Detection

The matching accuracy of automated face recognition systems has significantly improved over the past decade [98]. Indeed, challenges related to variations in pose, illumination and expression (PIE) have been identified and addressed by advanced algorithms that allow for unconstrained face recognition in diverse applications [98]. In spite of these advancements, there are still several factors that continue to challenge the performance of face recognition systems. These include factors related to aging [99], plastic surgery [100], and spoofing [101]. In a recent paper, Dantcheva et al. [5] demonstrated the negative impact of facial makeup on the matching performance of four face recognition algorithms. Their experiments suggested a significant decrease in matching accuracy when comparing facial images before and after the application of cosmetics. The use of makeup as a face alteration method poses a significant challenge to biometric systems, since it represents a simple, non-permanent, and cost effective way of confounding the system. Further, the use of makeup is socially acceptable in many parts of the world. Thus, detecting the presence of makeup in a face image can benefit face recognition systems from the perspective of both security (by flagging face spoofing or obfuscation attempts\(^1\)) and recognition accuracy (by facilitating the application of makeup-specific preprocessing routines). Additionally, automated age estimation and aesthetic prediction methods can utilize knowledge about the presence of makeup to refine their outputs.

\(^1\)Spoofing entails the use of makeup to look like another person. Obfuscation entails the use of makeup to mask one’s own identity.
In this work, we design a method to detect facial makeup in unconstrained face images. Given a face image, the proposed method first extracts a set of features based on shape, color and texture. This feature set is then used by a classifier to detect the presence or absence of makeup in the input face image. Experiments are conducted on two challenging and unconstrained datasets containing images of female subjects. The datasets include variations in facial pose, illumination, expression, and image resolution. Further, we use the output of the makeup detector to selectively pre-process face images prior to matching makeup images against no-makeup images. The proposed approach is observed to improve the matching performance of face recognition.

The rest of the chapter is organized as follows. Sections 4.1 and 4.2 describe the visual impact of makeup on the face and introduce the proposed makeup detection method. Section 4.3 introduces the databases that were assembled for this study. Section 5.6 presents experiments validating the effectiveness of the proposed method in detecting makeup. Section 4.5 introduces a face recognition scheme that exploits knowledge of the presence of makeup to selectively pre-process face images prior to matching. Finally, Section 4.6 concludes the chapter and discusses future directions.

### 4.1 Makeup Detection

Facial makeup is commonly used to enhance the aesthetics of a face, although it can also be used for concealing scars, moles and tattoos. In a recent British poll of 2,000 women\(^2\), more than half the subjects reported wearing makeup every day, with almost two thirds not leaving the house without makeup. A market research report\(^3\) indicates that the sales volume for makeup in the United States was 3.6 Billion in 2011 - a 9% increase from 2010. This suggests that the use of makeup is widespread and has become a daily necessity for many. While makeup consumers are predominantly female, the beauty market has been increasingly producing products geared toward a male clientele.

\(^2\)http://www.superdrug.com/content/ebiz/superdrug/stry/cgq1300799243/survey_release - jp.pdf

Table 4.1: Examples of face altering makeup items.

<table>
<thead>
<tr>
<th>Face region</th>
<th>Related makeup item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eye region</td>
<td>kohl, mascara, eye shadow, false eyelashes, eyebrow pencils, creams, waxes, gels and powders</td>
</tr>
<tr>
<td>Lip region</td>
<td>lipstick, lip gloss, liner, plumper, balm</td>
</tr>
<tr>
<td>Global skin</td>
<td>concealer, foundation, face powder, rouge, blush or blusher, contour</td>
</tr>
<tr>
<td>appearance</td>
<td>powder/creams, highlight, bronzer</td>
</tr>
</tbody>
</table>

Different types of makeup can be applied to different regions of the face. Table 4.1 gives a few examples. Makeup can fall under two categories:

- **Light makeup** (see Fig. 4.1(a)): The makeup cannot be easily perceived, since the applied colors correspond to natural skin, lip and eye colors.

- **Heavy makeup** (see Fig. 4.1(b)): The makeup is clearly perceptible (e.g. red or dark lips, strongly accentuated eyes).

The notion of light or heavy makeup does not necessarily relate to the number of makeup products that were used, but rather to the difference in facial appearance before and after applying makeup (Fig. 4.1).

Figure 4.1: The before and after makeup images of two subjects. The after makeup images exhibit color, shape and texture changes in the eye and mouth regions.

The aesthetic effects induced by makeup are a consequence of perceptual changes in facial appearance, which can be attributed to altered facial feature shapes due to contouring,
contrast changes in the mouth and eye region, and refined skin texture and color, as can be seen in Fig. 4.1. From a 2D image processing perspective we note that makeup can change the shape, texture and color information of global and local facial features. Therefore, it is essential to utilize both global and local information when detecting the presence of makeup.

4.1.1 Colorspace

One of the key aspects of the proposed approach is the choice of colorspace used to process the images. Based on visual assessment of makeup images in various colorspaces such as RGB (Red/Green/Blue), Lab (Luminance/Chromatic Components), and HSV (Hue/Saturation/Value), we decided to use the HSV colorspace. As can be seen in Fig. 4.2, information pertaining to makeup can be better discerned in the saturation channel of the HSV colorspace. The HSV color space is a non-linear transform of the RGB space and is given by ([102]):

\[
H = \arctan\left(\frac{\sqrt{3}(G-B)}{R-G + (R-B)}\right), \quad S = 1 - \min\{R,G,B\}, \quad V = \frac{R+G+B}{3}.
\]

![Figure 4.2: Visualizing a face as individual channels in the RGB (top) and HSV (bottom) colorspaces.](image)

4.1.2 Proposed Method

To the best of our knowledge, the only work related to automatic makeup detection is a very recent study by Varshovi [103], which was tested on 120 images of 21 frontal, neutral
expression female subjects and obtained classification accuracies of 90.62% for eye-shadow detection, 93.33% for lip-stick detection and 52.5% for liquid foundation detection. The study explored texture and color features for makeup cues.

We pose the problem of makeup detection as a two-class pattern classification problem. The makeup detector has the following components: (a) Face detection and landmark localization; (b) Face normalization; (c) ROI Extraction; (d) Feature Extraction; (e) Feature Classification (see Fig. 4.3). The AdaBoost face detector in OpenCV is used to provide an approximate location and scale of the face in the input image. Feature landmarks are then estimated within the facial region based on the method in [104] which employs a generative model for the landmark points and a discriminative model for the landmark appearance. The generative model is a Gaussian Mixture Model (GMM) that characterizes the joint probability distribution of landmark positions. The discriminative model consists of Haar-like filters and an AdaBoost classifier for locating and characterizing the appearance of each landmark. This is followed by face cropping and alignment based on the detected eye landmarks (estimated from the positions of left and right corner eye landmarks). For further processing, we consider this cropped facial area, as well as three specific regions of interest (ROIs): the regions around the left eye, the right eye and the mouth. Next, a set of shape, color and texture features are extracted from the face and ROIs (only color features are extracted from ROIs at this time), and a trained classifier is used to classify the extracted features into one of two classes: makeup or no-makeup. The proposed framework is illustrated in Fig. 4.3 and explained in detail below.

4.1.3 ROI Detection

After face detection and landmark localization, we geometrically normalize the face images using an affine transformation in order to remove variations due to scale and pose. All normalized face images are cropped and resized to a dimension of $150 \times 130$ pixels. Then the three ROIs are localized at pre-defined locations in the resized image and have the following dimensions: Left eye ROI: $52 \times 52$; Right eye ROI: $52 \times 52$; Mouth ROI: $56 \times 62$. Examples of these ROIs can be seen in Fig. 4.4.
Figure 4.3: Proposed framework for automatic facial makeup detection.

Figure 4.4: Examples of eye and mouth ROIs of 5 subjects. Top row: Without makeup. Bottom row: With makeup.
4.1.4 Feature Extraction

The proposed features for makeup detection are based on shape, texture and color descriptors. The choice of features was based on the following observations: (a) Visually, the dominant impact of makeup is on the color attributes of a facial image. Therefore, color-based features are used. (b) Since local shape and texture information are impacted by makeup, a set of Gabor filters are used to extract shape and texture information across different spatial scales and filter orientations. (c) Makeup can alter small-scale features in faces. Therefore, the Local Binary Pattern (LBP) operator is used to capture micro-pattern details of the facial image.

Below, we give details about the extracted features that are based on by Zhu et al. [105].

Color Descriptor

Color is a prominent low-level visual feature that can be used to describe images [102]. To extract color-based features, we first tessellate each ROI into $5 \times 5$ non-overlapping blocks and then compute color moments within every block (Fig. 4.5(a) and 4.5(c)). Let $I_{x,y}$ denote an image pixel at $(x,y)$ within a block in one of the channels. If $N$ is the total number of pixels, then the first order moment (mean) is calculated as $\rho = \sum_{x,y} \frac{1}{N} I_{x,y}$; the second order moment (standard deviation) as $\sigma = \sqrt{\frac{1}{N} \sum_{x,y} (I_{x,y} - \rho)^2}$; and the third order moment (skewness) as $\gamma = \sqrt[3]{\frac{1}{N} \sum_{x,y} (I_{x,y} - \rho)^3}$. These features are extracted from all 3 channels resulting in a 225-dimensional feature vector. To extract color moments from the entire face image, the face is partitioned into 9 non-overlapping block regions, resulting in an 81-dimensional color feature vector, as illustrated in Fig. 4.5(b) and 4.5(d).

Shape Descriptor

Three types of shape descriptors were used to extract additional features from the entire face. These are described below.

The first descriptor is based on Gabor wavelets and the same parameter settings specified in [77] were adopted. The size of each Gabor kernel was $64 \times 64$. Upon convolving the input face with the set of Gabor filters, we obtain 40 image outputs. We then calculate the mean,
Figure 4.5: The tessellation scheme used for extracting color-based features.

We utilize a second shape descriptor known as GIST, which was originally designed by Torralba and Oliva [106] for scene structure representation. It first applies pre-filtering to reduce illumination variations thereby preventing some local image regions from dominating the energy spectrum. The pre-filtering can be denoted as follows:

$$I'(x, y) = \frac{I(x, y) \times h(x, y)}{\epsilon + \sqrt{[I(x, y) \times h(x, y)]^2 \times g(x, y)}},$$  \hspace{1cm} (4.1)$$

where $I(x, y)$ is the input, $g(x, y)$ is a low-pass Gaussian filter and $h(x, y) = 1 - g(x, y)$ is the corresponding high-pass filter. Discrete Fourier Transform (DFT) is then applied to a set of Gabor filtered images (4 scales and 8 orientations) and the resultant image is divided into blocks by a $4 \times 4$ grid, from which the mean moment is extracted from each block. This results in a GIST feature vector of length $4 \times 4 \times 32 = 512$.

The third shape descriptor is based on edge information. Since the application of eye and mouth makeup enhances the local edge structure (corner and contour), an edge orientation
histogram is computed. A Canny edge detector is first applied to obtain the edge map, from
which an edge orientation histogram (EOH) [105] is extracted based on a 37-bin quantization
of edge orientation values (see Fig. 4.6).

![Image of original and edge images with edge histogram]

Figure 4.6: Examples of edge orientation histograms for the same subject with and without
makeup. The application of makeup typically increases edge information around the eyes
and the mouth.

Textual Descriptor

The LBP texture descriptor [23] is used to characterize micro-patterns or micro-structures
in the face image by binarizing local neighborhoods based on the differences in pixel intensity
between the center pixel and neighborhood pixels, and converting the resulting binary string
into a decimal value. Uniform LBP patterns (refers to those binary patterns that have at
most two bitwise transitions from 0 to 1 or 1 to 0) are extracted, resulting in a 59-bin
histogram feature vector (58 out of 256 patterns are uniform when a neighborhood size of 8
is used). Uniformity is an important aspect as it characterizes micro-features such as lines,
edges and corners, which are enhanced by the application of makeup.

The overall dimension of the feature vector, which integrates the color, shape and texture
features from both face and ROIs, is 1484. Separately, the dimensionalities of the face feature
vector and the ROI feature vectors are 809 and 675, respectively (see Table 4.2). Each feature
dimension is normalized to zero mean and unit variance.
Table 4.2: The dimensionality of features used in makeup detection.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Feature</th>
<th>Face-Dim</th>
<th>ROI-Dim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color</td>
<td>Moments</td>
<td>81</td>
<td>225 × 3</td>
</tr>
<tr>
<td>Shape</td>
<td>Gabor</td>
<td>120</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>GIST</td>
<td>512</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>EOH</td>
<td>37</td>
<td>-</td>
</tr>
<tr>
<td>Texture</td>
<td>LBP</td>
<td>59</td>
<td>-</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>809</strong></td>
<td><strong>675</strong></td>
</tr>
</tbody>
</table>

### 4.2 Classification

A pattern classifier, trained on labeled data, is used to classify the feature vector into one of two classes: “makeup” or “no-makeup”. We utilized the SVM [35] and Adaboost [87] classifiers in this work.

**SVM:** Support Vector Machine (SVM) searches for a linear boundary that maximizes the margin between two classes of patterns by solving the following optimization problem:

\[
\min_{w,\varepsilon} \left\{ \frac{1}{2}||w||^2 + C \sum_{i=1}^{N} \varepsilon_i \right\},
\]

subject to the constraint: \(y_i(w^T \cdot \phi(x_i) + b) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0\). Here, \(b\) is the bias and \(w\) is the weight vector, \((x_i, y_i)\) is the labeled \(i^{th}\) training sample, \(\varepsilon_i\) is a variable introduced to control the trade off between a large margin and a small error penalty, \(C\) is a constant and \(N\) is the total number of training samples. The Gaussian RBF kernel is used (to compute \(\phi\)) and defined as: \(k(x_i, x_j) = exp(-\gamma||x_i - x_j||^2)\), \(\gamma > 0\). The kernel is related to the transform \(\phi\) by the equation \(k(x_i, x_j) = \phi(x_i)^T \cdot \phi(x_j)\). The optimum values for \(C\) and the kernel parameter \(\gamma\) are obtained by a grid-search of the parameter space based on the training set.

**Adaboost:** The principle of Adaboost is to combine multiple weak classifiers to form a single strong classifier as \(y(x) = \sum_{t=1}^{T} \alpha_t h_t(x)\), where \(h_t(x)\) refers to the weak classifiers operating on the input feature vector \(x\), \(T\) is the number of weak classifiers, \(\alpha_t\) is the

---

A number of other classifiers were also experimented with. SVM and AdaBoost resulted in the best performance and are reported here.
corresponding weight for each weak classifier and \( y(x) \) is the classification output. In this work, for every pair of feature values \((f_i, f_j)\) in the feature vector \(x\), five types of weak binary classifiers are defined:

\[
h_t(x) \equiv g_k(f_i, f_j) = 1, \quad \text{if } d(f_i - f_j) > t_k, \quad (4.3)
\]

where \( k = 1 \ldots 5, t_1 = 0, t_2 = 5, t_3 = 10, t_4 = 25 \) and \( t_5 = 50 \). By changing the inequality sign from \( > \) to \( < \), another five types of weak classifiers are generated, resulting in a total of 10 types of weak classifiers. Since \( g_k \) is non-commutative, \( g_k(f_i, f_j) \neq g_k(f_j, f_i) \), the total number of weak classifiers for a pair of features \( f_i \) and \( f_j \) is 20.

Each pair-wise comparison results in a binary value, which is used as a weak-classifier by the Adaboost algorithm. For the 1484 dimensional feature vector, Adaboost will generate a set of \( T = \binom{1484}{2} \times 20 = 22,007,720 \) weak classifiers. For the 809 and 675 dimensional feature vectors extracted from face and ROIs, respectively, the number of weak classifiers are \( T = \binom{809}{2} \times 20 = 6,536,720 \) and \( T = \binom{675}{2} \times 20 = 4,549,500 \). After performing feature selection and weighting (for estimating \( \alpha_t \)) based on the classical Viola-Jones scheme, a total of 1000 weak classifiers are retained in each case.

### 4.3 Makeup Databases

#### 4.3.1 YouTube Makeup Database (YMU)

In this study, we utilized the database introduced by Dantcheva et al. [5] which contains the before and after makeup images of 151 Caucasian female subjects taken from YouTube makeup tutorials (99 subjects were used in their work in [5]). Examples are shown in Fig. 4.7 (after face cropping and alignment). For a majority of the subjects there are four shots per subject - two shots before the application of makeup and two shots after the application of makeup. For some subjects, there is either only one shot or three shots each before and after the application of makeup. The total number of images in the dataset is 600, with 300 makeup images and 300 no-makeup images. We note that the degree of makeup in this database varies from subtle to heavy. The database is relatively unconstrained, exhibiting variations in facial expression, pose and resolution.
Figure 4.7: Facial images showing variations in pose, illumination, expression and resolution from the YMU database [5]. The corresponding eye and mouth ROIs are shown in Fig. 4.4.

4.3.2 Makeup in the Wild Database (MIW)

In addition to the aforementioned dataset, we assembled another database of 154 images (77 with makeup, and 77 without makeup) corresponding to 125 subjects. Since the images are obtained from the Internet, we refer to this database as Makeup in the “Wild”. A few examples are shown in Fig. 4.8\(^5\). The purpose of using this database is to evaluate the generalization capability of the proposed makeup face detector where the training is performed using the YMU database and testing is done on the MIW database.

In both the databases, an image is labeled as “makeup” even if cosmetic details are present in only a portion of the face.

4.4 Experiments

In order to evaluate the performance of the proposed makeup detector, we employ a 5-fold cross-validation scheme. Here, the YMU dataset is divided into 5 folds with approximately 30 subjects in each fold. 4 folds are used for training the makeup detector, and the remaining fold is used for testing it. This is repeated 5 times. **Note that the subjects in the training set are not present in the test set.** The performance of the makeup detector is reported using two metrics: (a) Classification Rate (CR): The perf-

\(^5\)Images from the MIW database are available in the authors’ webpage: [http://www.antitza.com/makeup-datasets.html](http://www.antitza.com/makeup-datasets.html)
Figure 4.8: Sample images from the MIW database. Images are collected from the Internet. Top row shows images without makeup and the bottom row shows images with makeup. Note the unconstrained nature of the images.

4.4.1 Makeup Detection

In this section, we evaluate the performance of the proposed makeup detection system on the YouTube database. First, we investigate the performances of individual ROIs for makeup detection. As presented in Table 4.3, when extracting features from the entire face region, a classification rate of 87.25% using SVM was obtained. The left and right eye ROIs achieve classification rates of 81.71% and 80.68%, respectively. The mouth ROI has the lowest classification rate of 58.94%. This could be due to the inability of the color moments to capture the homogeneous region created by the application of lipstick. The SVM classification results for individual face feature sets are as follows: color moments: 77.62%; Gabor: 62.08%; GIST: 86.79%; EOH: 56.68%; LBP: 50.78%. When fusing the three 225-dimensional feature vectors corresponding to the individual ROIs, the classification percentage of makeup and no-makeup images that are correctly classified by the detector; (b) Receiver Operating Characteristic (ROC) curve: Here, the true positive rate (TPR: the percentage of “makeup” images that are correctly classified as “makeup”) is plotted as a function of the false positive rate (FPR: the percentage of “no-makeup” images that are incorrectly classified as “makeup”).
The classification rate for each trial in the 5-fold cross-validation experiment is reported in Table 4.4. Here, the Adaboost classifier obtains an average accuracy of 89.94 ± 1.60%, which is slightly lower than the SVM-based classifier (91.20 ± 0.56%).

Next, the proposed makeup detector that is trained on the YMU database (all 600 images), is tested on the MIW database. Sample outputs are presented in Fig. 4.10. The face detector failed in 22 of the 154 images. A classification rate of 95.45% for SVM and 92.21% for Adaboost was obtained. At 1% FPR, a TPR of 93.51% and 84.42% was obtained.
Table 4.3: The classification rates of the SVM-based and Adaboost-based makeup detector on the YMU database.

<table>
<thead>
<tr>
<th>ROI</th>
<th>SVM (%)</th>
<th>Adaboost (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entire Face</td>
<td>87.25 ± 1.91</td>
<td>88.98 ± 3.54</td>
</tr>
<tr>
<td>Left eye</td>
<td>81.71 ± 4.67</td>
<td>75.72 ± 1.99</td>
</tr>
<tr>
<td>Right eye</td>
<td>80.68 ± 3.08</td>
<td>79.89 ± 4.22</td>
</tr>
<tr>
<td>Mouth</td>
<td>58.94 ± 3.47</td>
<td>57.46 ± 5.94</td>
</tr>
<tr>
<td>Left eye + Right eye + Mouth</td>
<td>87.62 ± 2.01</td>
<td>85.83 ± 3.84</td>
</tr>
</tbody>
</table>

Table 4.4: Classification Rates of the SVM and Adaboost classifiers on the YMU database. The numbers in parentheses indicate the number of “no-makeup” and “makeup” images in each trial.

<table>
<thead>
<tr>
<th>Trial</th>
<th>Train</th>
<th>Test</th>
<th>SVM (%)</th>
<th>Adaboost (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>487 (243/244)</td>
<td>113 (57/56)</td>
<td>92.04</td>
<td>91.15</td>
</tr>
<tr>
<td>2</td>
<td>473 (237/236)</td>
<td>127 (63/64)</td>
<td>90.55</td>
<td>87.40</td>
</tr>
<tr>
<td>3</td>
<td>487 (244/243)</td>
<td>113 (56/57)</td>
<td>91.15</td>
<td>90.27</td>
</tr>
<tr>
<td>4</td>
<td>457 (228/229)</td>
<td>143 (72/71)</td>
<td>90.91</td>
<td>89.51</td>
</tr>
<tr>
<td>5</td>
<td>496 (248/248)</td>
<td>104 (52/52)</td>
<td>91.35</td>
<td>91.35</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td>91.20</td>
<td>89.94</td>
</tr>
</tbody>
</table>

for SVM and Adaboost, respectively. The corresponding ROC curves are shown in Fig. 4.9(f). This confirms the generalization ability of the proposed approach.

Experiments were conducted using Matlab R2009a on a 32 bit windows operating system with Intel Core i7-2600s CPU at 2.80GHz and 3.16GB RAM. The makeup detector processes a face image in 0.78 seconds.

4.5 Application in Face Recognition

In this section, we discuss the use of the proposed SVM-based makeup detector in the context of face recognition. In [5], the authors showed that the recognition performance of face matchers decreases when matching makeup images ($M$) against their no-make-up
counterparts ($N$). In order to address this issue, we devise a pre-processing routine. The idea here is to suppress the effect of makeup by utilizing a photometric normalization routine along with a blurring operator that smoothens the edge-like features induced by makeup. Specifically, if one of the two images to be matched is deemed to have makeup and the other is deemed to have no makeup, then both images are photometrically normalized using the Multiscale Self Quotient Image (MSQI) technique before they are input to the matcher.

The self-quotient image, $Q$, of image $I$ is defined as:

$$Q(x,y) = \frac{I(x,y)}{\hat{I}(x,y)} = \frac{\rho_w(x,y)n(x,y)s}{G_k * \rho_w(x,y)n(x,y)s},$$

where $\rho_w(x,y)$ is the albedo of the facial surface, $n$ is the surface normal, $s$ is the lighting reflection, $G_k$ is the weighted Gaussian smoothing filter and $k$ is the size of the kernel. In this work, four different kernel sizes were used (multi-scale): $3 \times 3, 5 \times 5, 11 \times 11, 15 \times 15$. The output image is the summation of the 4 filtered images. The corresponding sigma values used by the Gaussian filter were 1, 1.2, 1.4 and 1.6, respectively.

To test the efficacy of the scheme, we again use the YMU database and adopt the same 5-fold cross-validation scheme for evaluating performance. The Multi-Scale LBP (MSLBP) method is used for encoding and matching face images [108]. The MSLBP operator is very similar to the LBP operator (see Section 4.1.4) but with two main differences: (a) the binary pattern of a pixel is computed by comparing the mean values of sub-blocks; (b) the binary

---

6http://luks.fe.uni-lj.si/sl/osebje/vitomir/face_tools/INFace/index.html
pattern is computed at multiple scales and over a dense grid with spacing of 10 pixels. For a specific scale $s$, the size of the sub-block considered is $\frac{2}{3} \times \frac{2}{3}$. We consider 4 different scales in this work: 3, 9, 15, 21. For each scale a uniform LBP histogram is generated and the concatenated histogram values across the 4 scales serves as a feature vector for the face. Two such feature vectors are compared using the Histogram Intersection Distance to generate a match score. For each of the 5 trials we report results on the 3 matching scenarios suggested in [5]: no-makeup vs no-makeup images ($N$ vs $N$); makeup vs makeup images ($M$ vs $M$); makeup vs no-makeup images ($M$ vs $N$). In Table 5, the face verification rate is reported at a False Accept Rate (FAR) of 1%. “Aggregate” refers to the computation of verification results after pooling the match scores from all 5 trials for each scenario. It has to be noted that the application of the pre-processing routine increases the verification performance for the $M$ vs $N$ case, without impacting the accuracy of the $M$ vs $M$ and $N$ vs $N$ cases.

If pre-processing is applied to all image pairs before matching, then the corresponding verification results for $M$ vs $N$, $M$ vs $M$ and $N$ vs $N$ are 54.78%, 85.43% and 92.72%, respectively. While these results are comparable to the ones reported in Table 5, note that selective pre-processing as proposed in this work would (a) detect the presence of makeup and (b) avoid applying the pre-processing routine to all image pairs. The former is potentially useful information in identifying spoofing/obfuscation attempts. It must be noted that these are preliminary results, but do convey the applicability of the proposed makeup detector in face recognition.

### 4.6 Summary

In this chapter, we proposed an automated makeup detector for unconstrained facial images. The proposed detector utilizes shape, texture and color features extracted from the entire face, as well as facial subregions, to determine the presence of makeup. Experiments conducted on two unconstrained face datasets resulted in makeup detection rates of up to 93.5% (at 1% false positive rate) and overall classification rates of up to 95.45%. The output of the makeup detector was then used to perform adaptive pre-processing in the context of face recognition. Experimental results indicate that applying the proposed pre-processing
Table 4.5: Face verification performance (%) at a FAR of 1% before and after (B/A) applying the proposed face pre-processing scheme. The pre-processing scheme is invoked only when one image is deemed to have makeup and the other image is deemed to be without makeup by the proposed makeup detector. Column 3 highlights the improvement in verification results for the $M$ vs $N$ case.

<table>
<thead>
<tr>
<th>Trial</th>
<th>$M$ vs $N$</th>
<th>Increase</th>
<th>$M$ vs $M$</th>
<th>$N$ vs $N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>56.25/65.55</td>
<td>9.30</td>
<td>92.86/92.86</td>
<td>96.43/96.43</td>
</tr>
<tr>
<td>2</td>
<td>52.75/55.64</td>
<td>2.89</td>
<td>73.44/80.47</td>
<td>87.69/87.76</td>
</tr>
<tr>
<td>3</td>
<td>48.54/54.00</td>
<td>5.46</td>
<td>83.33/83.33</td>
<td>89.29/89.29</td>
</tr>
<tr>
<td>4</td>
<td>45.55/49.23</td>
<td>3.68</td>
<td>80.00/80.00</td>
<td>92.97/95.74</td>
</tr>
<tr>
<td>5</td>
<td>54.34/56.35</td>
<td>2.01</td>
<td>88.46/88.46</td>
<td>95.73/96.15</td>
</tr>
<tr>
<td>Aggregate</td>
<td>48.88/54.10</td>
<td>5.22</td>
<td>84.70/86.05</td>
<td>92.72/92.72</td>
</tr>
</tbody>
</table>

routine can improve the recognition accuracy of face matchers when matching makeup images against no-make-up images. However, more work is necessary in this regard. Future work will involve improving the performance of the makeup detector and exploring methods to remove artifacts introduced by the application of makeup. Specifically, we are interested in the problem of determining the degree of makeup applied to the face - this will have benefits in obfuscation/spoofing scenarios. Further, we will test the proposed method on datasets that include male subjects. Finally, the makeup detector can be used to refine the output of automatic age estimation and beauty assessment algorithms that may also be impacted by the application of makeup.
Chapter 5

Makeup-Robust Face Recognition

In this Chapter, we focus on the problem of matching face images before and after the application of makeup. These images are not acquired in a controlled environment, and hence considered as makeup in the wild. This problem is especially significant since makeup is a commonly used modifier of facial appearance. Thus, researchers in biometrics and cognitive psychology [74] are interested in understanding the effect of this modifier on face recognition.

5.1 Motivation and Related work

To date, there is limited scientific literature on addressing the challenge of make-up induced changes. Chen et al. [6] presented an automated makeup detection approach, that was used to adaptively modify images prior to performing face recognition. Hu et al. [109] used canonical correlation analysis (CCA) along with a support vector machine (SVM) classifier to facilitate the matching of before-makeup and after-makeup images. Guo et al. [52] learned the mapping between features extracted from patches in the before- and after-makeup images in order to minimize the disparity between the images to be matched. The mapping was learned using CCA, rCCA (regularized CCA) and Partial Least Squares (PLS) methods. While mapping-based methods have been shown to be effective, they have two main limitations. First, the mapping between before-makeup and after-makeup facial images can be complex, spatially variant and nonlinear. Therefore, it is insufficient to learn a single mapping in order to describe the complex relationship between before-makeup and after-makeup
samples [110]. Second, CCA and PLS methods have a tendency to overfit the training data and thus do not generalize well on unseen subjects [111].

In order to overcome these problems, we propose to use an ensemble learning scheme to generate multiple common semi-random subspaces for before-makeup and after-makeup samples, instead of two separate subspaces. In random subspace methods, a set of multiple low-dimensional subspaces are generated by randomly sampling feature vectors in the original high-dimensional space [112]. It has been proven to be effective in various tasks of face recognition [112, 113, 64, 114]. For instance, Wang and Tang [112] proposed the use of Random Subspace Linear Discriminant Analysis (RS-LDA) for face recognition by randomly sampling eigenfaces. Zhu et al. [113] randomly sampled features on local image regions to construct a set of base classifiers. Li et al. [114] extended the RS-LDA method to address the age-invariant face recognition problem. Zhang and Guo [115] utilized a feature selection module to construct random subspaces which are used for age estimation. Klare and Jain [64] adapted the RS-LDA method and proposed a heterogeneous face matcher for matching near-infrared images against visible images. The motivation for using a random subspace method are as follows [116]: (a) a learning algorithm can be viewed as searching for the best classifier in a space populated by different weak classifiers; (b) many weak classifiers are considered to be equally favorable when given a finite amount of training data; (c) averaging these individual classifiers can better approximate the true classifier. Therefore, a random subspace method can be used to generate multiple common subspaces, where each subspace contains a small portion of discriminative information pertaining to the identity. At the same time, by randomly selecting different patches as the input to each subspace-based classifier, the overfitting issue is avoided [112].

5.2 Proposed Method

In this work, a patch-based ensemble learning scheme for face recognition in the presence of makeup is proposed (see Figure 5.1). Given a face image, the proposed method first tessellates the image into patches and then applies multiple feature descriptors to each patch based on Local Gradient Gabor Pattern (LGGP) [81], Histogram of Gabor Ordinal Ratio
Measures (HGORM) and Densely Sampled Local Binary Pattern (DS-LBP). These descriptors capture both global (LGGP and HGORM) and local (DS-LBP) information. Next, a weight learning scheme based on Fisher’s separation criteria [117] is utilized to rank the significance of each patch. Then, a semi-random sampling method is used to select patches and construct multiple subspaces for each descriptor. Finally, Collaborative-based Representation Classifiers (CRC) and Sparse-based Representation Classifiers (SRC) are utilized in these subspaces resulting in an ensemble of classifiers for each descriptor. The scores generated by the classifiers are then fused using the sum-rule. The proposed method involves two levels of information fusion: the fusion of subspace classifiers corresponding to individual descriptors, and the fusion of matching scores generated by all descriptors. The rationale for the proposed method are as follows: (1) A single descriptor is not sufficient enough to describe a face image; (2) The prior knowledge about which patch is impacted by makeup is unknown; (3) Semi-random sampling can increase the probability of selecting patches that are not impacted by makeup. The proposed framework is illustrated in Figure 5.1.

Our approach to match two face images of the same person, acquired before and after the application of makeup, differs from previously published works with RS-LDA [113, 114, 64]. In the work of [114] and [113], the modified RS-LDA method is not specifically designed to handle heterogenous face recognition. In [64], the patch sampling procedure is performed across different feature descriptors (SIFT and LBP), while our patch sampling is performed within the same feature descriptor.

Contributions:

1. We propose an ensemble framework for a face matcher that is robust to the application of makeup. The approach utilizes multiple subspaces corresponding to three different feature descriptors and multiple image patches. A combination of sparse and collaborative classifiers are used in these subspaces.

2. We introduce a new feature descriptor (HGORM), which is an extension of HGOM [118] and LGGP [81] features.

3. We propose a sampling scheme, which utilizes weight information from each patch to
guide patch selection, instead of pure random sampling.

4. The proposed matcher is designed to perform well in other face recognition scenarios also, thereby underscoring its potential as a general purpose face matcher.

Figure 5.1: Proposed framework for matching after-makeup images with before-makeup images. During the training phase, for each feature descriptor, a pool of patches are extracted, followed by weight learning, patch sampling and random subspace construction. In the testing phase, patches from an input image are projected onto the learned random subspace. A combination of SRC and CRC classifiers are used to compare feature vectors in these subspaces and generate a match score. This process is repeated for each descriptor and the matching scores corresponding to individual feature descriptors are fused to generate the final similarity score.

5.3 Feature Descriptors

The design of effective face image descriptors is considered to be crucial in face recognition. In this work, the patches in each face image are represented using LGGP, HGORM
and DS-LBP descriptors. Compared to Dense HOG and Dense LBP used in [64], LGGP and HGORM are derived from Gabor-filtered images and have demonstrated to be more discriminative based on empirical investigation [81]. Let \( I \in \mathbb{R}^{128 \times 128} \) denote a face image that is either a before-makeup sample or an after-makeup sample, and let \( f \) denote a feature extractor. We now describe the three basic feature descriptors used in the framework.

**Gabor Filters:** A Gabor filter can be mathematically defined as follows [77]:

\[
\varphi_{\mu, \upsilon}(z) = \frac{||k_{\mu, \upsilon}||^2}{\sigma^2} e^{-\frac{||k_{\mu, \upsilon}||^2 ||z||^2}{2\sigma^2}} \left[e^{ik_{\mu, \upsilon}z} - e^{-\frac{z^2}{2}}\right],
\]

(5.1)

where \( \mu \) and \( \upsilon \) denote the orientation and scale of the Gabor filters, respectively. \( z \) denotes the pixel position and \( ||\cdot|| \) denotes the norm operator [117]. The wave vector \( k_{\mu, \upsilon} \) is given by \( k_{\mu, \upsilon} = k_{\upsilon} e^{i\phi_{\mu}} \), where \( k_{\upsilon} = k_{\text{max}} / s_{\upsilon} \) and \( \phi_{\mu} = \pi \mu / 8 \). Here, \( k_{\text{max}} \) is the maximum frequency and \( s \) is the spacing factor between kernels in the frequency domain. The Gabor response of an image is obtained by performing the convolution of the input image with Gabor kernels: \( G_{\mu, \upsilon}(z) = I(z) * \varphi_{\mu, \upsilon}(z) \). The complex Gabor response has two parts: the real part \( \Re_{\mu, \upsilon}(z) \) and the imaginary part \( \Im_{\mu, \upsilon}(z) \). Accordingly, the Gabor magnitude \( A_{\mu, \upsilon}(z) \) and phase \( \theta_{\mu, \upsilon}(z) \) can be computed as [119]:

\[
A_{\mu, \upsilon}(z) = \sqrt{\Re_{\mu, \upsilon}(z)^2 + \Im_{\mu, \upsilon}(z)^2},
\]

(5.2)

and

\[
\theta_{\mu, \upsilon}(z) = \arctan(\Im_{\mu, \upsilon}(z) / \Re_{\mu, \upsilon}(z)).
\]

(5.3)

Both Gabor magnitude and phase responses have been proven to be useful in face recognition [117, 120]. Since Gabor responses contain highly correlated and redundant information, it is essential to further encode such responses. It has been suggested that there are four different types of measurements from coarse to fine: nominal, ordinal, interval, and ratio measures [121]. To code Gabor responses, we utilize both ordinal and ratio measures to develop robust feature descriptors. The process of encoding Gabor-filtered images involves the following steps: (1) apply multi-orientation (eight orientations) and multi-scale (five scales) Gabor filters on the input face image; (2) derive either ratio or ordinal measures from the magnitude and phase components of the resulting images; (3) extract statistical distributions of these measures based on local histograms.
5.3.1 Local Gabor Gradient Pattern (LGGP)

To code Gabor magnitude responses, we use a gradient descriptor defined as [122]:

\[
\xi(x_c) = \arctan \left( \beta \cdot \frac{N_v}{N_h + \lambda} \right),
\]

where \( N_v \) and \( N_h \) are the image gradients to be computed along vertical and horizontal directions, respectively. Here, the two directions are orthogonal to each other. The arctangent function (arctan), along with parameters \( \beta \) and \( \lambda \), is used to prevent the output from increasing or decreasing too quickly [122]. Let \( \gamma_c \) define the intensity value of center pixel in a rectangle surrounded by neighbors equally sampled from \( \gamma_0 \) to \( \gamma_{R-1} \), where \( R \) is the neighborhood size. The gradients can now be computed as:

\[
N_v = \gamma_{\text{mod}(i+4,R)} - \gamma_i,
\]

\[
N_h = \gamma_{\text{mod}(i+6,R)} - \gamma_{\text{mod}(i+2,R)}.
\]

Here, the modulo operator is denoted by \( \text{mod} \) and \( i \) is the index for the neighborhood pixel. In our implementation, we use \( R = 8 \), \( \beta = 3 \) and \( \lambda = 1 \times 10^{-7} \). To generate LGGP features, each gradient-encoded Gabor image is divided into non-overlapping patches, and histogram information is extracted from each patch. The number of patches in each image is 64, where each patch size is \( 16 \times 16 \). The number of histogram bins is 16. LGGP feature extraction is denoted as \( f_G(I) = \{g_1, g_2, \ldots, g_P\} \), where \( P \) is the number of total patches and \( g_p \in \mathbb{R}^{16} \).

5.3.2 Histogram of Gabor Ordinal Ratio Measures (HGORM)

To code Gabor phase responses, we use the Ordinal Measure (OM) [123, 121]. OM compares two different regions to determine which one has a larger value (e.g., mean). For instance, if region A has a larger value than region B, then the resulting code is 1, otherwise it is 0. Such a measure is used to encode the qualitative relationship between different regions. The advantages of using ordinal measures for image representation have been established in palmprint recognition [123], iris recognition [121] and face recognition [118]. To perform ordinal feature extraction, one simple approach is to compute the weighted average of dissociated image regions. This can be accomplished by the process of ordinal filtering.
An ordinal filter consists of multiple positive and negative lobes, as illustrated in Figure 5.2. Here, we use multi-lobe differential filters (MLDF) [118] to extract ordinal features. The positive and negative lobes are represented by Gaussian filters. MLDF can be mathematically expressed as,

\[
MLDF = C_p \sum_{i=1}^{N_p} \frac{1}{\sqrt{2\pi\delta_{pi}}} \exp\left[-\frac{(z - \mu_{pi})^2}{2\delta_{pi}^2}\right] - C_n \sum_{j=1}^{N_n} \frac{1}{\sqrt{2\pi\delta_{nj}}} \exp\left[-\frac{(z - \mu_{nj})^2}{2\delta_{nj}^2}\right],
\]

where \( z \) denotes the pixel position, \( \mu \) and \( \delta \) denote the central position and the scale of a 2D Gaussian filter, respectively. \( N_p \) is the number of positive lobes and \( N_n \) is the number of negative lobes. \( C_p \) and \( C_n \) are the constant coefficients, used to ensure that the output of MLDF is zero, i.e., \( C_p N_p = C_n N_n \). MLDF is a type of differential bandpass filter. It is flexible in terms of types of lobes, spatial configuration of lobes, and number of lobes. An example of MLDF filters is shown in Figure 5.2.

![MLDF Filters](image)

Figure 5.2: Illustration of ordinal filters at different distances and orientations (horizontal, vertical). Positive and negative lobes are arranged in various configurations in terms of number of lobes and orientations between lobes.

Unlike the work of [118], we also consider the ratio measure in addition to the ordinal measure. First, we construct a horizontal di-lobe ordinal filter and a vertical di-lobe ordinal filter (see Figure 5.3). Then, we perform the ordinal filtering on the output of Gabor phase
responses. Finally, a ratio measure is used to compute the final representation:

\[ G = \arctan \left( \beta \cdot \frac{G_v}{G_h + \lambda} \right), \]  

where \( G_h \) is the convolution of horizontal di-lobe ordinal filter with the Gabor phase response and \( G_v \) is the convolution of vertical di-lobe ordinal filter with the Gabor phase response. The proposed feature representation is called Histogram of Gabor Ordinal Ratio Measures (HGORM). We use the following parameters in our work: the size of the ordinal filter is 21, the distance between positive and negative lobes is 3 pixels, \( \delta_{pi} = \delta_{nj} = \pi/2 \), \( N_p = N_n = 1 \), \( \beta = 3 \) and \( \lambda = 1 \times 10^{-7} \).

Figure 5.3: The horizontal and vertical ordinal filters used to compute the ratio measure. The distances between positive and negative lobes are same for these two filters.

HGORM can be considered as an extension of the LGGP descriptor that we previously developed [81]. The ratio measure used in HGORM is weighted by a Gaussian kernel, thereby making it more robust to noise. To generate HGORM features, each OM-encoded Gabor image is divided into non-overlapping patches, where histogram information is extracted from each patch. The number of patches in each image is 64, where each patch size is 16 \times 16. The number of histogram bins is 16. HGORM feature extraction is denoted as \( f_O(I) = \{ o_1, o_2, \ldots, o_P \} \), where \( P \) is the number of total patches and \( o_p \in \mathbb{R}^{16} \).

5.3.3 Densely Sampled Local Binary Pattern (DS-LBP)

The LBP texture descriptor [23] has been proven to be effective in capturing micro-patterns or micro-structures in the face image. It is calculated by binarizing local neighborhoods, based on the differences in pixel intensity between the center pixel and neighborhood
pixels, and converting the resulting binary string into a decimal value. In this work, uniform LBP patterns are extracted\(^1\) from the original image, resulting in a 59-bin histogram feature vector. Uniformity is an important characteristic, as it reflects micro-features such as lines, edges and corners, which are enhanced by the application of makeup. To generate DS-LBP features, each LBP coded image is divided into overlapping patches, and histogram information is extracted from each patch. The number of patches in each image is 256, where each patch size is 16 × 16. The number of histogram bins is 59. DS-LBP feature extraction is denoted as \(f_S(I) = \{s_1, s_2, \ldots, s_P\}\), where \(P\) is the number of total patches and \(s_p \in \mathbb{R}^{59}\).

All three feature descriptors are summarized in Table 5.1. The choice of feature descriptors was based on an empirical investigation of individual descriptor on the HFB database [81], and the consideration that the selected feature descriptors are complementary to each other.

Table 5.1: Summary of patch-based feature descriptors used in the proposed framework.

<table>
<thead>
<tr>
<th>Feature Types</th>
<th>LGGP</th>
<th>HGORM</th>
<th>DS-LBP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patch Size</td>
<td>16*16</td>
<td>16*16</td>
<td>16*16</td>
</tr>
<tr>
<td>Patch Tessellation</td>
<td>non-overlapping</td>
<td>non-overlapping</td>
<td>overlapping</td>
</tr>
<tr>
<td>Number of Patches</td>
<td>2,560</td>
<td>2,560</td>
<td>256</td>
</tr>
<tr>
<td>Number of Bins Per Patch</td>
<td>16</td>
<td>16</td>
<td>59</td>
</tr>
</tbody>
</table>

### 5.4 Semi-random Subspace LDA (SRS-LDA)

In this section, we describe the details of the proposed SRS-LDA method for matching after-makeup images to before-makeup images. Let \(I^B = \{I_i^B\}_{i=1}^c\) contain \(c\) classes of before-makeup samples, with each class \(I_i^B = \{I_i^{B,j}\}_{j=1}^{n_i}\) consisting of \(n_i\) samples \(I_i^{B,j}\) resulting in a total of \(N = \sum_{i=1}^c n_i\) before-makeup samples in the set. Similarly, after-makeup samples consisting of \(c\) classes are denoted as \(I^A = \{I_i^A\}_{i=1}^c\), where \(I_i^A = \{I_i^{A,j}\}_{j=1}^{m_i}\) and \(M = \sum_{i=1}^c m_i\).

In order to extract features corresponding to each descriptor, an image is divided into \(P\) patches (see Table 1). Therefore, a set of \(P\) feature vectors are extracted from a given

\(^1\)Uniform LBP patterns refers to those binary patterns that have at most two bitwise transitions from 0 to 1 or 1 to 0.
image. Let the set of feature vectors extracted from a before-makeup image be denoted as \(X_{i,j}^B = f(I_{i,j}^B)\). The \(p\)-th feature vector from \(X_{i,j}^B\) is denoted as \(X_{i,j}^B(p)\), where \(X_{i,j}^B(p) \in \mathbb{R}^{16}\) for LGGP and HGORM features, and \(X_{i,j}^B(p) \in \mathbb{R}^{59}\) for DS-LBP features. Similarly, \(X_{i,j}^A = f(I_{i,j}^A)\) is used to denote the set of feature vectors from an after-makeup image. Without loss of generality, \(X_{i,j}\) is used to denote the set of feature vectors extracted from \(I_{i,j}\), be it a before-makeup or an after-makeup image. For each of the three descriptors (LGGP, HGORM, and DS-LBP), the following procedure is adopted: training phase and testing phase (see Figure 5.1). The resultant matching scores are then fused from the three descriptors to make a final recognition decision.

5.4.1 Training Phase

Weight Learning: Before sampling patches, we assign a weight to each extracted patch and then rank the patches based on these weights. The weights are computed based on Fisher’s separation criterion [117]. Our assumption is that different facial regions may have different impact on face recognition across makeup, because makeup information is not uniformly distributed. For each patch \(p\) and its associated feature vector \(X_{i,j}(p)\), the mean of intra-class distance can be computed as:

\[
m_1(p) = \frac{1}{c} \sum_{i=1}^{c} \frac{2}{(l_i - 1)l_i} \sum_{j=1}^{l_i-1} \sum_{k=j+1}^{l_i} \phi(X_{i,j}(p), X_{i,k}(p)).
\]  

(5.9)

Here, \(\phi\) denotes the chi-squared distance between two feature vectors, \(l_i = n_i + m_i\) denotes the number of samples per class. The variance of intra-class distance can be computed as:

\[
\text{var}_1(p) = \sum_{i=1}^{c} \sum_{j=1}^{l_i-1} \sum_{k=j+1}^{l_i} (\phi(X_{i,j}(p), X_{i,k}(p)) - m_1(p))^2.
\]  

(5.10)

The mean of inter-class distance can be computed as:

\[
m_2(p) = \frac{2}{c(c-1)} \sum_{i=1}^{c-1} \sum_{j=i+1}^{c} \frac{1}{l_il_j} \sum_{j=1}^{l_i} \sum_{k=1}^{l_j} \phi(X_{i,j}(p), X_{i,k}(p)).
\]  

(5.11)

The variance of inter-class distance can be computed as:

\[
\text{var}_2(p) = \sum_{i=1}^{c-1} \sum_{j=i+1}^{c} \sum_{j=1}^{l_i} \sum_{k=1}^{l_j} (\phi(X_{i,j}(p), X_{i,k}(p)) - m_2(p))^2.
\]  

(5.12)
Then the weight of each patch $p$ is calculated as,

$$w(p) = \frac{(m_1(p) - m_2(p))^2}{\text{var}_1(p) + \text{var}_2(p)}.$$  \hspace{1cm} (5.13)

The patches are then sorted in descending order of their weights. This is used to guide the subsequent patch sampling process for each feature descriptor.

**Patch Sampling:** As stated earlier, multiple subspaces ($K$) are used to generate the ensemble of classifiers corresponding to each descriptor. Each subspace is constructed based on the semi-random\(^2\) sampling of the weighted patches. For creating the $k$-th subspace, $k = \{1, 2, \ldots K\}$, we sample $\alpha$ number of patches (without replacement) pertaining to a specific descriptor from $X_{i,j}^B$ and $X_{i,j}^A$, where $i = \{1, \ldots, c\}$, $j = \{1, \ldots, n_i\}$, and $\alpha \in \{1, \ldots, P\}$, to obtain $x_{i,j}^B \in \mathbb{R}^{D \times 1}$ and $x_{i,j}^A \in \mathbb{R}^{D \times 1}$. Here, $x_{i,j}^B$ and $x_{i,j}^A$ are obtained by concatenating all the $\alpha$ feature vectors into a single vector representation, where $D = \alpha \times d$ and $d$ is the feature dimension for each patch (see Table 5.1). Overall, 60% of $\alpha$ are selected from the first half of $P$ and 40% of $\alpha$ are selected from the remaining half. This ensures that more important patches have a higher chance to be selected.

**Subspace Construction:** Since the dimensionality of $D$ is usually much higher than the number of samples, a feature dimension reduction is performed to reduce computational time as well as to avoid the small sample size problem [112]. A common way to reduce the feature dimension is by applying Principle Component Analysis (PCA). PCA seeks to find the projection space, which can best reconstruct original vectors. To find such a subspace, mean vectors $\mu^B$ and $\mu^A$ are computed from before-makeup and after-makeup sampled feature vectors, respectively:

$$\mu^B = \frac{1}{N} \sum_{i=1}^{c} \sum_{j=1}^{n_i} x_{i,j}^B,$$

$$\mu^A = \frac{1}{M} \sum_{i=1}^{c} \sum_{j=1}^{m_i} x_{i,j}^A.$$  \hspace{1cm} (5.14)

An overall mean vector can be computed as $\mu = \frac{1}{N+M} \sum_{i=1}^{c} \sum_{j=1}^{n_i+m_i} x_{i,j}$. The entire covariance matrix can be computed as:

$$S = \frac{1}{N+M} \sum_{i=1}^{c} \sum_{j=1}^{n_i+m_i} (x_{i,j} - \mu)(x_{i,j} - \mu)^T.$$  \hspace{1cm} (5.14)

We can now compute eigenvectors $W_E$ from the covariance matrix $S$:

$$SW_E = \lambda W_E.$$  \hspace{1cm} (5.15)

\(^2\)Here, the term semi-random is used to indicate that the probability of selecting a patch is related to its weight.
After generating $W_E$, the before-makeup and after-makeup samples can be projected into the new subspace as:

$$ y^B_{i,j} = W^T_E (x^B_{i,j} - \mu^B), \tag{5.16} $$

and

$$ y^A_{i,j} = W^T_E (x^A_{i,j} - \mu^A). \tag{5.17} $$

$y^B_{i,j}$ and $y^A_{i,j}$ are the projected feature vectors after PCA for before-makeup and after-makeup samples, respectively. The number of eigenvectors used is $\min(N - c, M - c)$. We use both before-makeup and after-makeup feature vectors to compute the between-class scatter and within-class scatter matrices. This ensures that the learned feature representation is less sensitive to makeup changes. The mean class vector for $i$-th subject when constructing the $k$-th subspace for a feature descriptor is calculated using both before-makeup ($y^B_{i,j}$) and after-makeup ($y^A_{i,j}$) projected vectors:

$$ \mu^{(k)}_i = \frac{1}{n_i + m_i} \left( \sum_{j=1}^{n_i} y^B_{i,j} + \sum_{j=1}^{m_i} y^A_{i,j} \right). \tag{5.18} $$

Then the between-class scatter matrix can be computed as,

$$ S_B^{(k)} = \sum_{i=1}^{c} (\mu^{(k)}_i - \mu^{(k)})(\mu^{(k)}_i - \mu^{(k)})^T. \tag{5.19} $$

where $\mu^{(k)} = \frac{1}{c} \sum_{i=1}^{c} \mu^{(k)}_i$. The within-class scatter matrix can be computed as,

$$ S_W^{(k)} = \sum_{i=1}^{c} \sum_{j=1}^{n_i} (y^B_{i,j} - \mu^{(k)}_i)(y^B_{i,j} - \mu^{(k)}_i)^T + \sum_{i=1}^{c} \sum_{j=1}^{m_i} (y^A_{i,j} - \mu^{(k)}_i)(y^A_{i,j} - \mu^{(k)}_i)^T. \tag{5.20} $$

The objective of LDA is to seek the optimal projection $W_F$, which can maximize the ratio between determinant of the between-class scatter matrix and determinant of the within-class scatter matrix. The optimization problem is defined as,

$$ W_F^{(k)} = \arg \max_{W_F} \frac{W_F^T S_B^{(k)} W_F}{W_F^T S_W^{(k)} W_F}. \tag{5.21} $$

This is equivalent to solving the generalized eigenvalue problem of $S_B^{(k)} \psi^{(k)} = \lambda^{(k)} S_W^{(k)} \psi^{(k)}$, where $k = \{1, 2, \ldots, K\}$. The output of the training phase is $\mu^B, \mu^A, W_F^{(k)}$ and $W_E$ for each $k$ (i.e., subspace).
5.4.2 Testing Phase

In the testing phase, the after-makeup images of a subject in a sequestered test set are treated as probes and compared with before-makeup images that are treated as gallery images. Let $X_{i,j} = f(I_{i,j})$ denote the set of feature vectors extracted from $I_{i,j}$, where $I_{i,j}$ is either a before-makeup image or an after-makeup image from test samples. The same set of $\alpha$ patches are selected from $\{X_{i,j}(p) : X_{i,j}(p) \in X_{i,j}, 1 < p < P\}$, and concatenated to form a single feature vector $x_{i,j}^B$ or $x_{i,j}^A$.

**Subspace Projection:** For each derived subspace $k = \{1, 2, \ldots, K\}$, the representation for test samples of before-makeup and after-makeup are obtained as follows:

$$y_{i,j}^B = W_k^T W_k^T (x_{i,j}^B - \mu^B), \quad (5.22)$$

and

$$y_{i,j}^A = W_k^T W_k^T (x_{i,j}^A - \mu^A), \quad (5.23)$$

where $y_{i,j}^B$ and $y_{i,j}^A$ are the final projected feature vectors for before-makeup and after-makeup test samples, respectively. In case the makeup information is unknown, a makeup detection scheme [6] can be employed to make the distinction. An overall mean vector $\mu$ can also be used for projection, resulting in a slight decrease in matching accuracy ($< 1\%$ verification rate).

**SRC and CRC classification:** As stated earlier, the before-makeup samples are used as gallery, and the after-makeup samples are used as probe. Let $Y^B = \{y_{i,1}^B, \ldots, y_{i,n_i^B}^B, \ldots, y_{i',1}^B, \ldots, y_{i'^{c'},1}^B, \ldots, y_{i'^{c'},n_i'^{c'}}^B\}$ denote the gallery feature vectors of before-makeup samples, where $c'$ is the number of subjects in the gallery and $n_i'$ is the number of samples for the $i$-th subject. Let $y_{i,j}^A$ denote an after-makeup probe sample. Distance scores can be computed by matching a probe sample against a gallery, thereby obtaining a similarity score between before-makeup and after-makeup samples. This is accomplished by using the principles of sparse and collaborative representation. Wright et al. [24] demonstrated the robustness of sparse representation based classification (SRC) to occlusions and noise. The application of makeup can be viewed as the addition of noise, i.e., pixel corruption. Zhang et al. [31] demonstrated that it is the collaborative representation based classification (CRC),
rather than just sparsity, that lends robustness to face recognition. Thus, the collaborative role of face images from other subjects in the classification process cannot be undermined. SRC and CRC have their own merits and can provide complementary information in classification [124]. Therefore, we develop two classifiers for each subspace - one based on SRC and the other on CRC - and combine their outputs. In this way, we exploit the complementary information provided by both of them.

The coefficient of projection of an after-makeup sample can be obtained using both $\ell_1$ (SRC) and $\ell_2$ (CRC) solutions:

$$\rho_1^{(k)} = \arg\min_{\rho} \|Y^B \rho - y_{i,j}^A\|_2^2 + \lambda_1 \|\rho\|_1,$$

and

$$\rho_2^{(k)} = (\left((Y^B)^T Y^B + \lambda_2 I\right)^{-1} (Y^B)^T (y_{i,j}^A)).$$

The $\ell_1$ solution is obtained using Least Angle Regression (LARS) algorithm [125]. LARS algorithm is a variant for solving the Lasso based on model selection. The dimension of both $\rho_1^{(k)}$ and $\rho_2^{(k)}$ is $N' = \sum_{i=1}^{c'} n_i'$. The coefficient vector for the $k$-th subspace is $\rho^{(k)} = \rho_1^{(k)} + \rho_2^{(k)}$. The coefficient vectors pertaining to all subspaces are summed together:

$$\rho = \sum_{k=1}^{K} \rho^{(k)}.$$

Here, $\rho$ is the final coefficient (score) vector generated when matching an after-makeup feature vector $y_{i,j}^A$ (probe) to a set of feature vectors $Y^B$ (gallery). The proposed framework fuses the output of three different feature descriptors, i.e., LGGP, HGORM, and DS-LBP. So there are three coefficient vectors: $\rho_G$, $\rho_O$ and $\rho_S$. These are combined to get a single coefficient vector $\rho_F = (\rho_G + \rho_O + \rho_S)/3$. The match score between the $r$-th entry in the gallery and the probe image corresponds to the $r$-th element in $\rho_F$.

5.5 Baseline Algorithms

The accuracy of the proposed SRS-LDA method is compared against several face recognition algorithms. They are used as baselines in our analysis\(^3\).

\(^3\)We also tested the PLS and CCA methods proposed in [52]. The performance due to CCA was worse than the other matchers and, hence, not included in this chapter.
Figure 5.4: An example showing how SRC is used for identity classification during the ensemble learning. CRC follows the same principle, except that coefficients are spread widely. $\rho^{(k)}$ denotes the coefficients generated for the $k$-th subspace of a particular descriptor. Horizontal axis refers to the number of gallery samples and vertical axis refers to the coefficients for each sample as computed in Eqn. (6.29). The red circle denotes the coefficient associated with the first gallery sample in each subspace. The identity of a probe sample is correctly inferred after summing all the coefficients in the red circle.
Commercial Off-The-Shelf (COTS) Systems: Three commercial face recognition systems were evaluated in this study. To anonymize results obtained by these commercial matchers, they are referred to as COTS-1, COTS-2, and COTS-3. These are three of the ten matchers that were extensively evaluated in the NIST-organized Multi-Biometrics Evaluation [92]. Hence, these matchers are representative of state-of-the-art performance in face recognition. Comparing the accuracy of the proposed method against leading COTS systems provides an unbiased and objective baseline.

OpenBR: OpenBR [80] is a publicly available toolkit for biometric recognition and evaluation. The default face recognition algorithm in OpenBR is developed based on the Spectrally Sampled Structural Subspaces Features (4SF) [80]. An input face image is represented by extracting histograms of local binary pattern (LBP) and scale-invariant feature transform (SIFT) features, computed on a dense grid of patches. The histograms from each patch are then projected onto a subspace generated using PCA, in order to obtain a feature vector. LDA is then applied on each random sample to learn the discriminative subspace. The efficacy of the OpenBR face matching algorithm is elaborated in [80].

Histogram of Monogenic Binary Pattern (HMBP): HMBP [126] is established by concatenating the histograms of monogenic binary pattern (MBP) from all subregions. MBP is computed based on the output from monogenic signal analysis. The magnitude, phase and orientation of a 2D image is derived and local histogram is built from each non-overlapping subregion. The number of bins is 512. The final feature dimension for HMBP is 153,600.

Partial Least Square (PLS): Partial least square (PLS) is a statistical learning technique originally proposed in the field of chemometrics [52] as an alternative to ordinary least square regression. It maps input vectors (regressors) and corresponding output vectors (responses) into a common feature space such that the covariance between the projected input and output vectors is maximized. The number of basis vectors is 70.

5.6 Experiments

The following experiments were designed with the primary goal of exploring the effectiveness of the SRS-LDA method in matching after-make up to before-make up face samples.
Chapter 5. Makeup-Robust Face Recognition

5.6.1 Makeup Datasets

We utilized the YMU-dataset consisting of 151 subjects, specifically Caucasian females, from YouTube makeup tutorials. Images of the subjects before and after the application of makeup were captured. There are four samples per subject: two samples before the application of makeup and two samples after the application of makeup. The makeup in these face images varies from subtle to heavy. The cosmetic alteration is mainly in the ocular area, where the eyes have been accentuated by diverse eye makeup products. Additional changes are on the quality of the skin, due to the application of foundation and change in lip color. The database is relatively unconstrained, exhibiting variations in facial expression, pose and resolution. Some examples of YMU dataset are shown in Figure 5.5. The face images are geometrically normalized using an affine transformation, based on the eye landmarks, in order to reduce variations due to scale and pose. All normalized face images are cropped and resized to a dimension of 128 × 128. Images were converted from RGB to grayscale.

![Example images](image)

(a) Before-makeup samples

(b) After-makeup samples

Figure 5.5: The example images after alignment and cropping. Substantial change in facial appearance is observed after the application of makeup.

In addition to the aforementioned dataset, we assembled another makeup dataset for the purpose of training the proposed face matcher. The training dataset consists of a subset of female subjects from the FAM database, a female Asian makeup dataset from Youtube.

---

4Available at: http://www.antitza.com/makeup-datasets.html
5Although there are other intra-class variations in this dataset, it was determined in [5] that the drop in matching accuracy was primarily due to the application of makeup.
and the entire MIAA dataset [127] (see Figure 5.6). The total number of samples in this training dataset is 796, corresponding to 398 subjects. Each subject here has one before-makeup and one after-makeup sample. We refer to this dataset as “T-makeup”. Since the T-makeup dataset has limited number of samples per subject, we use the facial symmetry property to generate mirrored face samples. In this way, the size of the training dataset is doubled, which helps in the construction of more robust subspaces. The following parameter values were used in the experiment: the number of subspaces, \( K \), for each descriptor is 75; \( \lambda_1 = 0.15, \lambda_2 = 0.1, \alpha = 180 \) for HGORM and LGGP, \( \alpha = 80 \) for LBP. The number of dimensions of the SRS-LDA feature vector is 220.

![Sample images from the T-makeup training dataset](image)

Figure 5.6: Sample images from the T-makeup training dataset. Top row shows images before the application of makeup and bottom row shows images after the application of makeup.

### 5.6.2 Experiment on the YMU Database

This section encompasses experiments performed to demonstrate the merits of SRS-LDA for face recognition with makeup variations. In order to evaluate the performance of the proposed face matcher, genuine and impostor scores were generated according to the following protocol:

- **Match \( B \) against \( B \) (\( B \) vs. \( B \)): Both the images to be compared are before-makeup samples.**

- **Match \( A \) against \( A \) (\( A \) vs. \( A \)): Both the images to be compared are after-makeup samples.**
Table 5.2: Equal Error Rates (%) corresponding to the eight face matchers and three matching scenarios (B vs. B: matching of before-make-up images, A vs. A: matching of after-make-up images, and A vs. B: one of the images is after-make-up, the other is before-make-up) on YMU database (151 subjects and 604 images).

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>B vs. B</th>
<th>A vs. A</th>
<th>A vs. B</th>
</tr>
</thead>
<tbody>
<tr>
<td>COTS-1</td>
<td>3.85</td>
<td>7.08</td>
<td>12.04</td>
</tr>
<tr>
<td>COTS-2</td>
<td>0.69</td>
<td><strong>1.33</strong></td>
<td>7.69</td>
</tr>
<tr>
<td>COTS-3</td>
<td>0.11</td>
<td>3.29</td>
<td>9.18</td>
</tr>
<tr>
<td>OpenBR</td>
<td>6.87</td>
<td>16.44</td>
<td>25.20</td>
</tr>
<tr>
<td>LGBP</td>
<td>5.35</td>
<td>8.77</td>
<td>19.71</td>
</tr>
<tr>
<td>LGGP</td>
<td>5.36</td>
<td>8.01</td>
<td>19.70</td>
</tr>
<tr>
<td>HMBP</td>
<td>6.25</td>
<td>10.87</td>
<td>21.54</td>
</tr>
<tr>
<td>Proposed</td>
<td><strong>0.62</strong></td>
<td>1.99</td>
<td><strong>7.59</strong></td>
</tr>
</tbody>
</table>

- Match A against B (A vs. B): One of the images to be compared is after-make-up sample while the other is before-make-up sample.

The EERs (Equal Error Rates) of the matching scenarios considered in the YMU database are summarized in Table 5.2. COTS-1, COTS-2 and COTS-3 are commercial face recognition software, which represent state-of-the-art performances in the task of face recognition. OpenBR [80], LGBP [117], LGGP [81] and HMBP [126] are recent face recognition algorithms proposed in the academic field. The assessed algorithms all have significantly higher EERs, when matching after-make-up to before-make-up samples. The EER of the proposed method for face matching scenario A vs. B is 7.59%. We have significantly reduced the EER from over 20% (see OpenBR, HMBP) to 7.59%. Further, the performance is better than all the three COTS matchers. The proposed method achieves the best EER for the A vs. B case. PLS⁶, on the other hand, obtains an EER of 23.91% on A vs. B. The PLS model was trained with the same feature descriptors as SRS-LDA.

We also considered fusing the proposed method with COTS⁷, to further improve the matching performance. Individual match scores generated by different matchers are normal-

---

⁶Code used: http://www.cs.umd.edu/~djacobs/pubs_files/PLS_Bases.m
⁷Fusing with COTS-1 results in poor performance, and is thus not used in the analysis.
ized based on min-max rule, followed by the simple sum rule. As can be seen from Figure 5.7 and Figure 5.8, the fused matchers significantly improve the face matching performance in terms of both EER and Genuine Accept Rate (GAR). It is apparent that the proposed method and COTS provide complementary information. As reported in Table 5.3, COTS1, COTS2 and COTS3 obtain EERs of 12.04%, 7.69%, and 9.18%, respectively. COTS1, COTS2 and COTS3 obtain GARs (GAR: 0.1% FAR) of 48.86%, 76.15%, and 58.48%, respectively. The proposed method achieves EER of 7.59% and GAR of 69.24%. When fusing the proposed method with COTS-2 and COTS-3, we obtain the best results: 83.61% GAR and 5.19% EER, which is better than the fusion of COTS-2 and COTS-3. This clearly indicates that commercial systems have further room for improvement and that the proposed method addresses this issue. It must be noted that the proposed method focuses only on robust feature extraction and matching; this is in contrast to end-to-end COTS matchers which have the advantage of many years of research and are, therefore, likely to have advanced pre-processing and post-processing routines. Inspite of this, the proposed method is observed to be very competitive.

Figure 5.7: Performance evaluation of proposed method and different COTS systems with reported ROC curves (EER: A vs. B) on YMU database.
Figure 5.8: Performance evaluation of proposed method and different COTS systems with reported ROC curves (GARs at different FAR: A vs. B) on YMU database. This is the semilogarithmic plot of Figure 5.7.

Table 5.3: The result of fusing the proposed method with different COTS systems on the YMU database. GAR performance at a FAR of 0.1% is reported.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>GAR (%)</th>
<th>EER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>COTS-1</td>
<td>48.86</td>
<td>12.04</td>
</tr>
<tr>
<td>COTS-2</td>
<td>76.15</td>
<td>7.69</td>
</tr>
<tr>
<td>COTS-3</td>
<td>58.48</td>
<td>9.18</td>
</tr>
<tr>
<td>Proposed</td>
<td>69.24</td>
<td>7.59</td>
</tr>
<tr>
<td>Proposed+COTS-2</td>
<td>79.88</td>
<td>5.98</td>
</tr>
<tr>
<td>Proposed+COTS-3</td>
<td>74.44</td>
<td>6.59</td>
</tr>
<tr>
<td>COTS-2+COTS-3</td>
<td>80.54</td>
<td>5.98</td>
</tr>
<tr>
<td>Proposed+COTS-2+COTS-3</td>
<td><strong>83.61</strong></td>
<td><strong>5.19</strong></td>
</tr>
</tbody>
</table>
Due to the use of semi-random sampling scheme\textsuperscript{8}, it is evident that the method is not deterministic. In spite of that, we demonstrate that the solution is rather stable. In order to verify the stability of the SRS-LDA method, we repeat the experiments 31 times and report the distributions of both EER and GAR. As seen from Figure 5.9, the EER values range from 6.92\% to 7.80\% and the GARs at a FAR of 0.1\% range from 67.35\% to 69.64\%. In our experiments, we use the median value to report the result\textsuperscript{9}.

![Figure 5.9: Demonstration (boxplot) of stability of the SRS-LDA method on YMU database: 7.52 ± 0.23 (EER), 68.51 ± 0.59 (GAR at FAR=0.1%).](image)

**Analysis of Individual Features**

In the proposed framework, we utilize three types of features (LGGP, HGORM and DS-LBP). However, the performance of individual feature descriptors vary (see Table 5.4). As seen in Table 5.4, if we use descriptor-based methods only, then the matching performance of A vs. B is very poor. LGGP\textsuperscript{10}, HGORM and DS-LBP obtain EERs of 20.48\%, 20.24\% and 19.65\%, respectively. These results are consistent with other descriptor-based methods reported in Table 5.2. This demonstrates the necessity to use an ensemble learning scheme, to further improve the performance. After adopting the SRS-LDA method, the matching performance increases significantly. The choice of these features (LGGP, HGORM and DS-

---

\textsuperscript{8}The semi-random sampling scheme resulted from weight learning improves GAR by approximately 4\% over random sampling on A vs. B.

\textsuperscript{9}A one-sided hypothesis test at the 5\% significance level indicates that the data comes from a population with a mean less than 7.69\% EER.

\textsuperscript{10}Result is reported with slightly different parameters as in Table 5.2.
Table 5.4: Performance of individual feature descriptors on YMU database. GAR performance at a FAR of 0.1% is reported.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>GAR (%)</th>
<th>EER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGGP</td>
<td>32.82</td>
<td>20.48</td>
</tr>
<tr>
<td>HGORM</td>
<td>37.99</td>
<td>20.24</td>
</tr>
<tr>
<td>DS-LBP</td>
<td>30.26</td>
<td>19.65</td>
</tr>
<tr>
<td>SRS-LDA+LGGP</td>
<td>57.99</td>
<td>11.01</td>
</tr>
<tr>
<td>SRS-LDA+HGORM</td>
<td>63.64</td>
<td>10.21</td>
</tr>
<tr>
<td>SRS-LDA+DS-LBP</td>
<td>58.06</td>
<td>11.35</td>
</tr>
</tbody>
</table>

LGBP) are based on empirical testing. We have also analyzed the matching performance of LGBP and HMBP. However, we cannot simply fuse all feature descriptors together due to implicit correlation between some of the feature descriptors. After rigorous investigation, we found that the fusion of LGGP, HGORM and DS-LBP gives the best result, thereby justifying the use of these features in the proposed framework. HGORM achieves the best overall performance among individual feature descriptors.

**Number of Subspaces**

An important parameter to be considered in SRS-LDA is the number of subspaces used ($K$). For this purpose, we conducted another experiment to analyze the convergence property of the proposed algorithm. We gradually increase the number of subspaces and compute the corresponding Rank-1 accuracies. As illustrated in Figure 5.10, the performance of the proposed algorithm first increases with the number of subspaces, and then stabilizes. HGORM reaches its peak in 26 iterations, while DS-LBP and LGGP reach their respective peaks after 43 and 54 iterations.

In our experiment, we simply set the number of subspaces to be 75. We notice that after this, the performance stabilizes. This illustrates that the proposed method is not very sensitive to the number of subspaces selected. This stabilization aspect is one of the characteristics of the proposed random subspace method.
Figure 5.10: The impact of number of subspaces on the SRS-LDA algorithm for different feature descriptors (YMU database).

Computational Complexity

The proposed SRS-LDA method is computationally efficient to meet the requirements of practical face recognition systems. Since the random subspaces generated by the ensemble learning algorithm are independent of each other, they can be processed in parallel, making them suitable for large scale face recognition. The training phase is done offline. Hence, only the feature extraction and classification of the test samples impact the computational time during real-time operation. The feature extraction processes (Eqn. (5.22) and Eqn. (5.23)) are simple linear operations and introduce very limited overhead. The computational complexity is indicated in Table 5.5.

Table 5.5: The computational time (seconds) for the proposed algorithm in terms of feature extraction and classification steps.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>LGGP</th>
<th>HGORM</th>
<th>DS-LBP</th>
<th>SRC</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (s)</td>
<td>4.18</td>
<td>3.00</td>
<td>0.09</td>
<td>0.54</td>
<td>0.19</td>
</tr>
</tbody>
</table>

Experiments were conducted using Matlab R2012b on a 64 bit windows operating system with Intel Core i7-2600 CPU at 3.40GHz and 8GB RAM\(^\text{11}\).

\(^\text{11}\)DS-LBP is implemented in MEX which makes it faster than pure MATLAB:
Table 5.6: Rank-1 accuracies (%) of different face matchers before and after adding FRGC mugshots to the gallery.

<table>
<thead>
<tr>
<th>Gallery →</th>
<th>YMU</th>
<th>YMU + FRGC</th>
<th>YMU + FRGC + MIW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>80.46</td>
<td>78.48</td>
<td>77.81</td>
</tr>
<tr>
<td>COTS-2</td>
<td>86.42</td>
<td>85.43</td>
<td>84.44</td>
</tr>
<tr>
<td>COTS-3</td>
<td>75.83</td>
<td>74.50</td>
<td>72.19</td>
</tr>
<tr>
<td>Proposed+COTS-2</td>
<td>88.41</td>
<td>86.42</td>
<td>85.43</td>
</tr>
<tr>
<td>Proposed+COTS-3</td>
<td>85.76</td>
<td>81.46</td>
<td>80.79</td>
</tr>
<tr>
<td>COTS-2+COTS-3</td>
<td>88.41</td>
<td>88.08</td>
<td>86.75</td>
</tr>
<tr>
<td>Proposed+COTS-2+COTS-3</td>
<td><strong>89.40</strong></td>
<td><strong>88.74</strong></td>
<td><strong>87.75</strong></td>
</tr>
</tbody>
</table>

5.6.3 Large-scale Identification Experiment

To demonstrate a practical face retrieval experiment, we augment the before-makeup samples in the gallery with a subset of images from the FRGC database [38], similar to the protocol specified in [128]. In this experiment, the after-makeup samples from the YMU database are used as probes. A subset of 10,000 (10K: 4574 females + 5426 males) mugshot images are selected from the FRGC database. So the gallery comprises of 10,302 images (302 before-makeup YMU images and 10K FRGC images). We use the exact same matching algorithm as described in Subsection 5.6.2. As seen in Table 5.6, the Rank-1 accuracy drops only marginally in the case of the proposed algorithm in spite of expanding the size of the gallery. This suggests that the proposed method is scalable over a large database.

Furthermore, we conduct another experiment where a set of 112 face images with makeup from the MIW dataset [6] are added to the gallery, resulting in a total of 10,414 gallery samples. Figure 5.11 shows the Cumulative Match Characteristic (CMC) curve, before and after adding the mugshots. The performance of the proposed algorithm is observed to be better than the commercial algorithms on the specified task. The performance is further improved after fusing the proposed method with the COTS matchers.


12The identities of these after-makeup samples in the gallery do not overlap with that of after-makeup samples in the probe.
Figure 5.11: The CMC curves of different face matchers before and after adding mugshots. YMU only: 302 after-makeup samples from YMU are used as probes and 302 before-makeup samples from YMU are used in the gallery; YMU + FRGC Mugshots: 302 after-makeup samples from YMU are used as probes, while 302 before-makeup samples from YMU and 10K FRGC mugshots are used in the gallery; YMU + FRGC Mugshots + MIW: 302 after-makeup samples from YMU are used as probes, while 302 before-makeup samples from YMU, 112 after-makeup samples from MIW [6], and 10K mugshots from FRGC are used in the gallery.
Examples of successful and unsuccessful matching can be found in Table 5.7. As illustrated in the examples of false matched pairs, the proposed algorithm sometimes inadvertently learns inter-class variations such as specific hair-styles and poses. Additionally, examples of successful and unsuccessful matching where (a) COTS-2 failed; (b) COTS-3 failed; (c) the proposed method succeeded; (d) and the fusion succeeded, are shown in Table 5.8.

Table 5.7: Examples of true and false matches at Rank-1. The selection of these matching pairs (A vs. B) is based on the face identification scenario.

<table>
<thead>
<tr>
<th>True Match Pairs</th>
<th>False Match Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="True Match Pairs" /></td>
<td><img src="image2" alt="False Match Pairs" /></td>
</tr>
</tbody>
</table>

Table 5.8: Examples of successful Rank-1 matches that demonstrate the effectiveness of fusion (Proposed+COTS-2+COTS-3).

<table>
<thead>
<tr>
<th>Probe</th>
<th>(a) COTS-2</th>
<th>(b) COTS-3</th>
<th>(c) Proposed</th>
<th>(d) Fusion</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image3" alt="Probe" /></td>
<td><img src="image4" alt="COTS-2" /></td>
<td><img src="image5" alt="COTS-3" /></td>
<td><img src="image6" alt="Proposed" /></td>
<td><img src="image7" alt="Fusion" /></td>
</tr>
</tbody>
</table>
5.6.4 Comparison against HFR Matcher

The proposed method is also evaluated against a state-of-the-art Heterogeneous Face Recognition (HFR) matcher that is used in [64] and [8]. The HFR matcher\(^{13}\) is a random subspace method that is constructed based on the random sampling of dense LBP (Dense-LBP) and dense SIFT features (Dense-SIFT) [8]. The classification is performed using the Nearest Neighbor Classifier (NNC) or the Sparse Representation Classifier (SRC). We reimplement the algorithm\(^{14}\) and set the following parameters: \(K = 75\) and \(\alpha = 180\). The matcher is trained on the T-makeup dataset and tested on the YMU makeup dataset. For NNC and SRC, the selected patches in each subspace are the same. As seen from Table 5.9, the proposed method is much better than the HFR matcher.

Table 5.9: Comparison against a HFR matcher [8]. The proposed method uses both SRC and CRC classifiers. GAR performance at a FAR of 0.1% is reported.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>GAR (%)</th>
<th>EER (%)</th>
<th>Rank-1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>69.24</td>
<td>7.59</td>
<td>80.46</td>
</tr>
<tr>
<td>HFR+NNC</td>
<td>39.80</td>
<td>16.39</td>
<td>68.21</td>
</tr>
<tr>
<td>HFR+SRC</td>
<td>46.36</td>
<td>19.52</td>
<td>72.52</td>
</tr>
</tbody>
</table>

5.6.5 Experiment on the FERET Database

In order to verify the generalization capability of the proposed SRS-LDA method, we used the same trained model from the T-makeup dataset and performed face matching on the FERET dataset\(^{15}\). We achieve Rank-1 accuracies of 96.57%, 90.72%, 79.07%, 74.36% on Fb, Fc, Dup1 and Dup2 subsets, respectively. The CMC curves are shown in Figure 5.12.

\(^{13}\)Dense-LBP and Dense-SIFT take about 0.09 and 0.16 seconds to extract features, respectively. SRC and NNC take about 0.07 and 0.04 seconds to compute the matching scores, respectively.

\(^{14}\)It is not an exact reimplementation and some details may vary.

\(^{15}\)To further validate that the proposed matcher learns makeup-invariant features, we use FERET subsets (Fa and Fb) as training and T-makeup dataset as testing. The obtained performances are: 28.04% EER, 15.75% GAR at 0.1% FAR and 28.39% Rank-1. This poor performance indicates the need for using a training set that has images with makeup.
corresponding EERs are 0.47%, 1.41%, 2.20% and 2.36%, respectively. The reported Rank-1 accuracies are comparable to many popular algorithms evaluated on this dataset [120, 126, 129]. To the best of our knowledge, the fused performance (Proposed+COTS-2+COTS-3) is one of the best results obtained on the FERET database [118].

![Graph showing identification rates](image)

Figure 5.12: Reported Rank-k identification rates on FERET using the proposed method. The T-makeup dataset is used to train the SRS-LDA face matcher.

The lower performance on Fc is due to the fact that the feature descriptor DS-LBP used by SRS-LDA is not robust to illumination changes. We do not apply any illumination preprocessing scheme in our proposed model. Our intention is not to claim that the SRS-LDA is superior to COTS in every aspect. Instead, we attempt to convey the idea that SRS-LDA method has the ability to be used as a general purpose face matcher although it has been designed for addressing the problem of makeup.

5.7 Discussions

In this section we summarize the main observations made from the experimental results.

- The impact of makeup on face recognition is significantly reduced by using the proposed SRS-LDA method. We obtain 7.59% EER, 69.24% GAR (0.1% FAR), and 80.46% Rank-1 accuracy on the YMU database.
Table 5.10: Rank-1 performance (%): comparison of the proposed method with other state-of-the-art commercial matchers on the FERET dataset.

<table>
<thead>
<tr>
<th></th>
<th>Fb</th>
<th>Fc</th>
<th>Dup1</th>
<th>Dup2</th>
</tr>
</thead>
<tbody>
<tr>
<td>COTS-1</td>
<td>98.33</td>
<td>99.48</td>
<td>86.01</td>
<td>81.62</td>
</tr>
<tr>
<td>COTS-2</td>
<td>98.58</td>
<td>100.0</td>
<td>93.21</td>
<td>91.03</td>
</tr>
<tr>
<td>COTS-3</td>
<td>99.16</td>
<td>98.97</td>
<td>91.97</td>
<td>93.16</td>
</tr>
<tr>
<td>Proposed</td>
<td>96.57</td>
<td>90.72</td>
<td>79.09</td>
<td>74.36</td>
</tr>
<tr>
<td>Proposed+COTS-2</td>
<td>99.08</td>
<td>100</td>
<td>94.18</td>
<td>92.31</td>
</tr>
<tr>
<td>Proposed+COTS-3</td>
<td>99.50</td>
<td>99.48</td>
<td>92.94</td>
<td>91.88</td>
</tr>
<tr>
<td>Proposed+COTS-2+COTS-3</td>
<td>99.58</td>
<td>100</td>
<td>96.26</td>
<td>95.73</td>
</tr>
<tr>
<td>COTS-2+COTS-3</td>
<td>99.67</td>
<td>100</td>
<td>96.54</td>
<td>96.15</td>
</tr>
</tbody>
</table>

• While the SRS-LDA method is designed to specifically learn feature invariance before and after the application of makeup, it performs reasonably well as a general purpose face matcher (see Appendix).

• SRS-LDA lends itself to parallel processing; further, SRS-LDA is flexible, where other types of feature descriptors can easily be incorporated into the framework.

• Fusion of SRS-LDA with COTS can further improve the matching performance. This clearly demonstrates that SRS-LDA can provide complementary information, which can be utilized to improve the matching accuracy of commercial matchers. We obtain 5.19% EER, 83.61% GAR (0.1% FAR), and 89.40% Rank-1 accuracy on the YMU database.

5.8 Summary

This Chapter presents a method for matching after-makeup images with their before-makeup counterparts. We provide extensive evidence that the proposed method is competitive and outperforms several state-of-the-art descriptor-based methods and commercial matchers. The proposed method uses a patch-based ensemble learning scheme, where mul-
multiple subspaces are generated for three different descriptors. The Fisher’s separation criteria is used to guide the patch sampling process prior to generating the subspaces. Both SRC and CRC classifiers are utilized for classification in the semi-random subspaces. The final output is the fusion of matching scores from individual descriptors. Experimental results on the YMU database demonstrate the effectiveness of the proposed method. The fusion of proposed method with COTS further improves matching accuracy.
Chapter 6

Cross-spectral Face Recognition

6.1 Local Gabor Gradient Pattern (LGGP)

Automated face recognition is a very active and challenging research topic in biometrics. In its simplest form, the problem of face recognition entails comparing two face images and determining if they are of the same face. Deducing an effective face representation or encoding scheme is still an open research problem. There are two main methods in the literature: learning-based methods [130, 129] and descriptor-based methods [23, 25]. Learning-based methods attempt to learn the concept of a face based on a set of training examples. The learned concept is then used to compare and match two face images. However, in some cases, such an approach might suffer from the generalization problem when coping with previously unseen images that bear little resemblance to the training set. On the other hand, descriptor-based methods do not explicitly employ a learning procedure. Instead, a feature set is extracted from the facial image and directly used when comparing it against another face image. The two approaches are often combined to improve recognition accuracy [130, 77] in face recognition.

Thus far, many descriptor-based methods have been introduced and successfully employed in face recognition. Ahonen et al. [23] used Local Binary Pattern (LBP) to extract micro-patterns from images and applied the descriptor to the face recognition task. Another popular descriptor is the Gabor wavelet, which has been widely used in face recognition due to the similarity of its responses to that of the visual cortex in the human vision system.
Figure 6.1: The general architecture considered in this work. An input image is subject to a Gabor Filter resulting in a Gabor Response (magnitude here). The Gabor Magnitude is encoded using different descriptors. The resulting coding values are represented as spatial histograms. The image is from the FERET database.

[131]. Liu et al. [77] proposed the use of augmented Gabor features, along with the Fisher Linear Discriminant (FLD) method to recognize faces. Tan et al. [130] fused both Gabor and LBP features to enhance face recognition performance. Recently, a serial combination of the Gabor and LBP descriptors was performed by employing LBP to encode Gabor transformed images. Zhang et al. [25] proposed a non-statistical model called Local Gabor Binary Pattern Histogram Sequence (LGBPHS or LGBP) to represent faces. A face descriptor was obtained by concatenating the local region histograms of the LBP-encoded Gabor magnitude patterns. The proposed method demonstrated the best matching accuracy on the FERET face database at the time of its publication. Since then, numerous techniques for combining Gabor filters and LBP descriptors have been presented in the face recognition literature [34, 132, 133, 129].

In contrast to the LGBP approach where the Gabor magnitude information is used, Gabor phases have been encoded using LBP to demonstrate their effectiveness in face recognition [34]. However, LBP is not the only way to encode the magnitude or phase information. Histogram of Gabor Phase Patterns (HGPP) [120] and Local Gabor XOR Patterns (LGXP) [119] have also been used to encode Gabor phase by using the XOR operator. Nicolo and Schmid [133] extended the use of Weber Local Descriptor (WLD) [122] to the problem of multispectral face recognition. In their work, WLD descriptor, along with LBP was used to encode both the Gabor magnitude and phase responses.
In this work, we introduce a simple descriptor for encoding the Gabor filter response. The proposed descriptor is based on the gradient of the Gabor filter response and is demonstrated to be useful not only for the problem of face recognition, but also for cross-spectral matching and soft biometric prediction.

Our work is an extension of the methods described in previous literature [25, 122, 133]. The basic idea is that the Gabor filter response of an image can be further enhanced or encoded with image descriptors before being utilized for feature extraction. Here, we summarize the main contributions of this work.

Our first contribution is to propose a new coding method and demonstrate its strength compared to other techniques. Inspired by recent work on illumination invariant face recognition, we propose to encode the gradient information of the Gabor filter response in order to achieve robustness against photometric variations. The proposed method achieves the best average accuracy on four probe sets of the FERET database among descriptor-based approaches.

Second, we demonstrate that combining the Gabor-based and LBP methods in a serial manner, as described in the coding framework (see Fig.6.1), is better than a parallel framework where the Gabor-based and LBP methods are applied independently to the image. This provides a good perspective on how to effectively utilize two of the arguably most successful descriptors [130, 77, 25].

Third, we evaluate the proposed face representation method in the context of cross-spectral (heterogeneous) face recognition and demonstrate the utility of such a descriptor in dealing with face images captured in different wavelength bands (Visible and Near-infrared).

6.1.1 Image Preprocessing

Face images captured from different modalities need to be preprocessed in order to help compensate for the appearance difference. This is accomplished by applying image filtering methods to extract texture features under varying illumination conditions [134]. These approaches seek to reduce modality gap based on image-formation models and are of great essence to enhance cross-spectral matching [66, 81]. According to the Lambertian model,
the image formation process is described as follows:

\[ I(x, y) = \rho_w(x, y)n(x, y)s, \]  

(6.1)

where \( \rho_w(x, y) \) is the albedo of the facial surface, \( n \) is the surface normal and \( s \) is the lighting reflection. The Lambertian model usually assumes that the term \( \rho_w(x, y) \) is constant across different lighting sources. However, since the lighting conditions under NIR and VIS spectra are not homogeneous, estimating an illumination invariant albedo \( \rho_w(x, y) \) under the Lambertian model for those two type of images is not possible [135]. Therefore, approaches based on the Lambertian model, such as self-quotient image and its variants, are not useful in our application. Since the reflectance is not a stable characteristic of facial features for images captured under the NIR and VIS spectra, the Retinex model also does not result in good performance. Only those normalization methods based on local appearance-based features (i.e., Retinex and DoG) result in better accuracy.

**Retinex Model:** The Retinex approach is based on the reflectance illumination model instead of the Lambertian model. It is an image enhancement algorithm [136] proposed to account for the lightness and color constancy of the dynamic range compression properties of the human vision system. It tries to compute the invariant property of reflectance ratio under varying illumination conditions [137, 107]. The Retinex model is described as follows:

\[ I(x, y) = R(x, y)L(x, y), \]  

(6.2)

where \( I(x, y) \) is the image, \( R(x, y) \) is the reflectance of the scene and \( L(x, y) \) is the lighting. The lighting is considered to be the low-frequency component of the image \( I(x, y) \), and is thus approximated as,

\[ L(x, y) = G(x, y) * I(x, y), \]  

(6.3)

where \( G(x, y) \) is a Gaussian filter and \( * \) denotes the convolution operator. The output of the Retinex approach is the image \( R(x, y) \) that is computed as,

\[ R(x, y) = \frac{I(x, y)}{L(x, y)} = \frac{I(x, y)}{G(x, y) * I(x, y)}. \]  

(6.4)

**Difference-of-Gaussian (DoG) Filtering:** Another type of normalization is proposed in [32], where the local image structures are enhanced. One of the key components in [32] is
the Difference-of-Gaussian (DoG) filtering, which can be computed as,

\[ D(x, y | \sigma_0, \sigma_1) = [G(x, y, \sigma_0) - G(x, y, \sigma_1)] * I(x, y). \] (6.5)

The symbol \(*\) is the convolution operator, and the Gaussian kernel function based on \(\sigma\) is,

\[ G(x, y, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x^2+y^2)}{2}\sigma^2}. \] (6.6)

This simple filtering scheme has the effect of subtracting two Gaussian filters.

We attempt to apply these image normalization techniques to reduce spectrum difference induced by sensors, leading to a more accurate description of features that represent identity.

### 6.1.2 Descriptors and Encoding Methods

The reason a coding procedure is adopted after the application of the Gabor filter to an input face image is to improve the robustness of the filter to changes in image characteristics. While the raw values of the Gabor filter response may contain redundant information, the purpose of the coding procedure is to not necessarily perform feature reduction, but to enhance the feature discrimination capability. The high dimensionality of the features is effectively reduced by employing the spatial histogram scheme in the subsequent stage. All descriptors have the same coded feature length, regardless of which coding method is used. Therefore, new descriptors can be easily developed and incorporated in this framework.

**Gabor Filters:** The Gabor wavelets are defined as follows [77]:

\[ \varphi_{\mu, \nu}(z) = \frac{|k_{\mu, \nu}|^2}{\sigma^2} e^{-\frac{|k_{\mu, \nu}|^2|z|^2}{2\sigma^2}} \left[ e^{ik_{\mu, \nu}z} - e^{-\frac{\sigma^2}{2}} \right], \] (6.7)

where \(\mu\) and \(\nu\) denote the orientation and scale of the Gabor kernels, \(z\) denotes the pixel position, i.e., \(z = (x, y)\) and \(|\cdot|\) denotes the norm operator [25]. The wave vector \(k_{\mu, \nu}\) is given by:

\[ k_{\mu, \nu} = k_{\nu} e^{i\phi_{\mu}}, \] (6.8)

where \(k_{\nu} = k_{\text{max}} / f^\nu\) and \(\phi_{\mu} = \pi \mu / 8\). Here, \(k_{\text{max}}\) is the maximum frequency and \(f\) is the spacing factor between kernels in the frequency domain. The Gabor wavelet representation of an image is obtained by convolving the input image with Gabor kernels:

\[ G_{u, v}(z) = I(z) * \varphi_{u, v}(z). \] (6.9)
The complex Gabor response has two parts, the real part \( \Re_{u,v}(z) \) and the imaginary part \( \Im_{u,v}(z) \). The Gabor magnitude \( A_{u,v}(z) \) and phase \( \theta_{u,v}(z) \) can be obtained as [119]:

\[
A_{u,v}(z) = \sqrt{\Re_{u,v}(z)^2 + \Im_{u,v}(z)^2},
\]

and

\[
\theta_{u,v}(z) = \arctan(\Im_{u,v}(z)/\Re_{u,v}(z)).
\]

**Weber Local Descriptor:** Weber Local Descriptor (WLD) is derived from Weber’s Law, which states that the ratio of the increment threshold to the background intensity is a constant [122]:

\[
\frac{\Delta I}{I} = k,
\]

where \( \Delta I \) is the increment threshold and \( I \) is the background intensity. \( k \) is a constant value, often known as the Weber fraction. The WLD descriptor can be described as [122]:

\[
WLD(x_c) = \arctan \left[ \sum_{i=0}^{P-1} \frac{x_i - x_c}{x_c} \right],
\]

where \( \arctan \) is the arctangent function that helps improve the robustness of WLD to noise. \( x_c \) is the center pixel surrounded by neighbors \( x_i \) equally sampled from \( x_0 \) to \( x_{P-1} \), where \( P \) is the neighborhood size. A slight modification of the approach leads to:

\[
WLD(x_c) = \arctan \left[ \alpha \sum_{i=0}^{P-1} \frac{x_i - x_c}{x_c + \lambda} \right].
\]

Here, \( \alpha \) is a factor to magnify or shrink the difference between neighbors. \( \lambda \) is a small constant value to avoid division by zero. The default values for \( \alpha \) and \( \lambda \) are 3 and \( 1 \times 10^{-7} \), respectively, in our experiments. When WLD is used to encode the Gabor filter response, the method is termed as Weber’s Local Gabor Pattern (WLGBP) [133].

### 6.1.3 Our Proposed Technique

As described in the previous subsection, WLD actually exploits the contrast information within an image patch, which can still be sensitive to illumination changes. It has been shown
that the ratio of change in intensity along the $y$ direction to that along the $x$ direction is approximately constant across different illumination changes [138]:

$$g = \frac{\partial I(x,y)/\partial y}{\partial I(x,y)/\partial x}, \quad (6.15)$$

based on the illumination model:

$$I(x,y) = R(x,y) \cdot L(x,y), \quad (6.16)$$

where $R(x,y)$ is the reflectance and $L(x,y)$ is the illuminance. This means:

$$I(x + \delta x, y) - I(x, y) = R(x + \delta x, y) \cdot L(x + \delta x, y) - R(x, y) \cdot L(x, y) \approx (R(x + \delta x, y) - R(x, y)) \cdot L(x, y), \quad (6.17)$$

assuming that $L(x+\delta x, y)$ and $L(x, y)$ are approximately smooth [138]. Taking the derivative of the above equation with respect to $\delta x$ will lead to $\frac{\partial I(x,y)}{\partial x} = \frac{\partial R(x,y)}{\partial x} L(x, y)$. Similarly, changing $\delta x$ to $\delta y$ in the derivation will result in $\frac{\partial I(x,y)}{\partial y} = \frac{\partial R(x,y)}{\partial y} L(x, y)$. Considering that the measurement of $R$ is an illumination insensitive measure, the ratio of gradient $y$ and gradient $x$ in Eqn. (6.15) is also an illumination insensitive measure [138]. The gradient descriptor can now be computed as [122]:

$$\xi(x_c) = \arctan \left( \alpha \cdot \frac{N_y}{N_x + \lambda} \right), \quad (6.18)$$

where $N_y$ and $N_x$ are the gradients to be computed along $y$ and $x$ directions, respectively. Here, the two directions are orthogonal to each other, depicting an orientation of $0^\circ$. The orientation can be changed by rotating $N_y$ and $N_x$ simultaneously. The parameters $\alpha$ and $\lambda$ are used to stabilize the gradient descriptor. We define $x_c$ as the center pixel in a rectangle surrounded by neighbors equally sampled from $x_0$ to $x_{P-1}$, where $P$ is the neighborhood size. The gradients can now be computed as:

$$N_y = x_{\text{mod}(i+4, P)} - x_i, \quad (6.19)$$

$$N_x = x_{\text{mod}(i+6, P)} - x_{\text{mod}(i+2, P)}. \quad (6.20)$$
In our implementation, we use $P = 8$, $\alpha = 3$ and $\lambda = 1 \times 10^{-7}$. $\mod$ is the modulo operator and $i$ is the index for the neighborhood pixel. Histogram features are extracted from the gradient-encoded Gabor-filtered images to generate the corresponding image signature. This proposed encoded feature is termed as Local Gradient Gabor Pattern (LGGP) (see Fig. 6.2(c)). The proposed gradient encoding method has the following properties:

- Unlike conventional LBP coding which generates a binary string first and then converts this sequence to a decimal value (LBP-like descriptor), the proposed coding method directly utilizes the intensity values to calculate the transformation without producing intermediate binary sequence values. Such a scheme can effectively avoid the loss of information during the conversion.

- Compared to WLD where contrast information is exploited, the proposed encoding method utilizes the gradient information which has been shown to be more effective against lighting variations [138, 139].

The WLD and Gradient descriptors are illustrated in Fig. 6.2. As can be observed here, the WLD descriptor captures prominent features while neglecting details. On the other hand, the Gradient descriptor captures more detailed structures in the facial images. A similar idea for encoding Gabor energy variations by extracting separate $x$ and $y$ gradient components of Gabor responses has been reported by Guo et al. [140]. However, our utilization of gradient information is different in the sense that the ratio of gradients is encoded for each pixel. Similar to LBP, the method can be further extended from basic rectangular structures (e.g., $3 \times 3$ or $5 \times 5$) to circular regions (see Fig. 6.3). We can also change the encoding orientation.

6.1.4 Evaluation on FERET Benchmark Datasets

The FERET face database was used in our evaluation. The original images were cropped based on eye coordinates and normalized to $128 \times 128$ pixels. The same standard gallery and probe sets as specified in the FERET evaluation protocol were used [1]. For the Gabor filters, the default parameters were set as follows: $\mu \in \{0, 1, \ldots, 7\}$, $\nu \in \{0, 1, \ldots, 4\}$, $\sigma = \pi$, $k_{max} = \pi/2$, $f = \sqrt{2}$, and the size of Gabor kernel is $31 \times 31$. For the feature extraction
(a) Without Gabor transformation  
(b) With Gabor transformation for $\mu = 7$ and $\nu = 4$

(c) LGGP Visualization

Figure 6.2: Examples of WLD and Gradient coding schemes. The original images are of the same subject from the FERET dataset under different illumination conditions.

Figure 6.3: An example showing how to compute the Gradient descriptor. $x_c$ is the center pixel surrounded by neighbors equally sampled from $x_0$ to $x_{P-1}$, where $P$ is the neighborhood size. $r$ is the defined radius of the neighborhood. Different parameters such as scale, orientation, and neighborhood size can be selected accordingly. The two given equations compute the descriptor at different orientations of $0^\circ$ and $45^\circ$, respectively.
part that involves computation of histograms in non-overlapping blocks, the block size is 16 × 16 and the number of histogram bins is 16. The dimension of the feature vector is, therefore, 8 × 8 × 16 × 40 = 40,960. These parameters were used in all the experiments, unless otherwise specified in the narrative below.

**Baseline Performance**

As we have discussed before, the coding step is very important as it is expected to generate a robust descriptor that is invariant to different image conditions. In order to show the effectiveness of such a coding mechanism, we compare the WLGBP, LGBP and LGGP methods against the baseline performance. The baseline is obtained without employing any coding methods, i.e., direct Gabor filter response is used, in conjunction with spatial histograms. Establishing a good baseline approach is critical to the objective comparison of different face recognition algorithms. As seen from Table 6.1, the LBP coding methods (LGBP_Mag and LGBP_Pha) exceed the baseline performance in all probe sets. For the Fc set, where illumination changes are presented, the LBP coding scheme improves the Rank-1 performance from 89.18% to 95.36% accuracy. For the Dup2 set across time lapse,

<table>
<thead>
<tr>
<th>Methods</th>
<th>Fb</th>
<th>Fc</th>
<th>Dup1</th>
<th>Dup2</th>
<th>Ave.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>93.05</td>
<td>89.18</td>
<td>60.25</td>
<td>61.97</td>
<td>76.11</td>
</tr>
<tr>
<td>WLGBP</td>
<td>98.16</td>
<td>94.85</td>
<td>70.22</td>
<td>66.24</td>
<td>82.37</td>
</tr>
<tr>
<td>LGBP_Mag</td>
<td>96.23</td>
<td>95.36</td>
<td>72.58</td>
<td>68.80</td>
<td>83.24</td>
</tr>
<tr>
<td>LGBP_Pha</td>
<td>96.82</td>
<td>96.39</td>
<td>76.45</td>
<td>73.93</td>
<td>85.90</td>
</tr>
<tr>
<td>LGGP_Mag</td>
<td>97.66</td>
<td><strong>96.91</strong></td>
<td><strong>77.70</strong></td>
<td><strong>79.06</strong></td>
<td><strong>87.83</strong></td>
</tr>
<tr>
<td>LGGP_Pha</td>
<td>96.32</td>
<td>96.39</td>
<td>77.42</td>
<td>74.79</td>
<td>86.23</td>
</tr>
<tr>
<td>Gabor</td>
<td>75.40</td>
<td>88.66</td>
<td>27.98</td>
<td>20.09</td>
<td>53.03</td>
</tr>
<tr>
<td>LBP</td>
<td>93.05</td>
<td>65.46</td>
<td>60.53</td>
<td>52.14</td>
<td>67.80</td>
</tr>
<tr>
<td>Gabor+LBP (F)</td>
<td>93.56</td>
<td>73.71</td>
<td>61.50</td>
<td>50.00</td>
<td>69.69</td>
</tr>
</tbody>
</table>
the performance is boosted from 61.97% to 68.80%. Such improvements in performance suggest that LBP-based coding indeed increases the robustness of the Gabor filter response to textural photometric changes. The WLD coding method (WLGBP) surpasses the baseline performance in all probe sets. It achieves the best Rank-1 accuracy of 98.16% on Fb set. However, it appears that WLGBP is still not robust enough to deal with challenging datasets such as Dup1 and Dup2. It is also interesting to observe that WLGBP results in higher recognition accuracy than LGBP when the matching is performed under well-constrained conditions (Fb set). In the Fc set, where illumination variations are presented, WLGBP results in slightly lower performance than LGBP. This also suggests that WLD is a bit sensitive to illumination changes. Our proposed gradient-based encoding method (LGGP) achieves the best overall performance of 87.83% when the Gabor magnitude is encoded and 86.23% when the Gabor phase response is encoded. It demonstrates robustness under difficult image characterisitics of illumination (Fc set with 96.91%) and time lapse variation (Dup2 set with 79.06%) as seen in Table 6.1.

**Comparison to State-of-The-Art Methods**

To further enhance the proposed LGGP feature sets, we use the Difference-of-Gaussian (DOG) normalization method proposed in Tan et al. [130] to handle variations in illumination conditions. Additionally, match scores generated using a multi-scale configuration (3 × 3 and 5 × 5) are combined together through sum rule to enhance the performance. The improved method is termed as multi-scale LGGP (MS-LGGP).

To show the effectiveness of MS-LGGP, we further compare the results against other published works (Table 6.2). These results are directly obtained from the respective original publications. As can be seen, the proposed method performs better than most existing descriptor-based methods, such as LBP [23], HMBP [82], MGCP [140] and HGPP [120], in all probe sets. It also obtains the best average accuracy among all the listed methods. It is worth mentioning that the implementation of LGBPHS in Zhang et al. [25] is different than ours due to the choice of parameters used. POEM-HS [141] employed Retina filter to preprocess the face images. There are many other papers that have results reported on the same database. However, they utilize a machine learning approach, which may not generalize
very well when new samples are introduced. Therefore, we are only interested in comparing our scheme against descriptor-based methods.

For completeness sake, we also report identification accuracy using the Cumulative Match Characteristic (CMC) curve that is similar to the protocol used in CSU face recognition evaluation [142]. Another benchmark algorithm based on V1-like features [131], which is a concatenation of locally-normalized, thresholded Gabor filtered images based on 16 orientations and 6 frequencies, is also included for reference. All of the Rank-k identification results are summarized in Fig. 6.4.

Table 6.2: Rank-1 performance (%): comparison of the proposed methods with other state-of-the-art descriptor-based methods on FERET.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Fb</th>
<th>Fc</th>
<th>Dup1</th>
<th>Dup2</th>
<th>Ave.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBP [23]</td>
<td>93.0</td>
<td>51.0</td>
<td>61.0</td>
<td>50.0</td>
<td>63.75</td>
</tr>
<tr>
<td>LGBPHS [25]</td>
<td>94.0</td>
<td>97.0</td>
<td>68.0</td>
<td>53.0</td>
<td>78.00</td>
</tr>
<tr>
<td>LGBP_Pha [34]</td>
<td>93.0</td>
<td>92.0</td>
<td>65.0</td>
<td>59.0</td>
<td>77.25</td>
</tr>
<tr>
<td>HMBP [82]</td>
<td>97.74</td>
<td>98.45</td>
<td>73.55</td>
<td>72.22</td>
<td>85.49</td>
</tr>
<tr>
<td>HGPP [120]</td>
<td>97.6</td>
<td>98.9</td>
<td>77.7</td>
<td>76.1</td>
<td>87.58</td>
</tr>
<tr>
<td>MGCP [140]</td>
<td>97.4</td>
<td>97.3</td>
<td>77.8</td>
<td>73.5</td>
<td>86.50</td>
</tr>
<tr>
<td>POEM-HS [141]</td>
<td>98.1</td>
<td>99.0</td>
<td>79.6</td>
<td>79.1</td>
<td>88.95</td>
</tr>
<tr>
<td>LGXP [119]</td>
<td>98.0</td>
<td>100</td>
<td>82.0</td>
<td>83.0</td>
<td>90.75</td>
</tr>
<tr>
<td>E-GV-LBP [129]</td>
<td>98.41</td>
<td>98.97</td>
<td>81.99</td>
<td>81.62</td>
<td>90.25</td>
</tr>
<tr>
<td>MS-LGGP</td>
<td>97.99</td>
<td>98.97</td>
<td><strong>83.93</strong></td>
<td>82.48</td>
<td><strong>90.84</strong></td>
</tr>
</tbody>
</table>

### 6.1.5 Evaluation on HFB Dataset

The use of Near-infrared (NIR) images for face recognition has become necessary especially in the context of a night-time environment where Visible (VIS) face images cannot be easily discerned [33]. Here, a common feature-based representation for both NIR images as well as VIS images is used, similar to Klare and Jain [143]. Unlike approaches from existing work [143, 63] where associations between heterogeneous data (VIS and NIR) are learned
Figure 6.4: Performance evaluation of different methods with the reported CMC curves.
from discriminant analysis in order to generate a common feature subspace, our solution is to directly derive a face image representation that is capable of compensating for the differences between two spectra without applying machine learning techniques. Our analysis shows that the gradient information extracted from both VIS and NIR is able to preserve the identity across different spectra to a certain extent.

The HFB face database [67] consists of 100 subjects, including 57 males and 43 females. There are 4 VIS and 4 NIR face images per subject. The variations in appearance between corresponding VIS and NIR samples are significant (see Fig. 6.5(a)), which are caused by differences in sensor and lighting characteristics [144]. Cross-spectral face recognition was observed to be difficult [143, 145]. One of the key issues, therefore, is to reduce the variability between these two type of images by applying an illumination normalization routine (see Fig. 6.5(b)). As a result, facial structures are well preserved after accounting for reflections, shadows and other variations across spectra. Furthermore, a carefully designed feature extraction framework is also critical to address the problem. We utilize the observation that the appearance of corresponding small regions would not exhibit large variations across different spectra. Therefore, a block size of $8 \times 8$ is chosen, instead of $16 \times 16$ in the experiments below. LGBP, GDBC [146], and LGGP methods achieve accuracies of 84.0%, 86.75% and 88.0%, respectively (Fig. 6.6). MS-LGGP obtains a slightly better performance of 88.25% compared to LGGP with 88.0%. A rank-1 accuracy of 90.75% (Fused) is obtained by the score-level fusion (sum rule) of LGBP and MS-LGGP methods.

![Images before DoG normalization](image1)

![Images after DoG normalization](image2)

(a) Images before DoG normalization  
(b) Images after DoG normalization

Figure 6.5: Samples of face images in visible (top) and near-infrared (bottom) spectra. Images are from HFB database and not co-registered.

In general, intra-spectral face matching (VIS-VIS and NIR-NIR) can achieve relatively
higher performance compared to cross-spectral (VIS-NIR) matching. For VIS-VIS matching (two images per subject were used as probe), we achieve Rank-1 accuracies of 100% and 98.5% for LGGP and LGBP, respectively. For NIR-NIR matching, we obtain Rank-1 accuracies of 99.5% and 95.0% for LGGP and LGBP, respectively. The other important observation is the decrease in verification performance of NIR-NIR matching (LGGP: 6.48% EER) compared to VIS-VIS matching (LGGP: 2.16% EER). This is possibly due to the fact that the identity information presented in the NIR image domain may not be as discriminative as in the VIS domain.

There are only a few papers that present results on direct matching between VIS and NIR on the HFB database. In the work of Li et al. [67], the best Rank-1 accuracy, based on a 70-subject training set and a 30-subject test set, does not exceed 50.0%. Recent work on a larger version of this database (200 subjects) [144] reported the best Rank-1 accuracy of 98.51% with 50 test subjects. However, their approach utilizes a learning-based scheme to select the best features. Though learning-based schemes can often significantly improve the matching performance, it can suffer from the generalization problem when a new dataset is introduced. A more realistic solution would be to construct a descriptor that can be flexibly incorporated in different scenarios, such as VIS-VIS and NIR-VIS matching, without
undergoing significant modifications.

6.2 Matching Remote Near-infrared to Visible

In this section, we focus on the face matching between VIS and NIR images, where VIS images are used as galleries and NIR images are used as probes. This problem is further complicated when NIR probe images are captured at long distances. These remote NIR images contain degradations from sensor noise, blur, low resolution and artifacts (see Figure 6.7). These variations present immediate concerns to existing HFR matchers [63, 9, 147]. We define Remote Heterogeneous Face Recognition (RHFR) when NIR probe images are captured at long distances (60-150m) from the cameras and VIS gallery images are captured in an indoor environment (1m). Hence, we need to address two problems simultaneously: (a) matching NIR face images against VIS face images (cross-spectral face recognition), and (b) matching low resolution face images against high resolution face images (cross-distance face recognition).

To date, there is limited work on addressing the challenge of RHFR. Maeng et al. [9] utilized Gaussian-filtered Scale Invariant Feature Transform (Gauss-SIFT) and RS-LDA method [143] for cross-spectral and cross-distance face matching, which was trained on HFB [148] and tested on LDHF-DB [9]. Their experiments revealed that cross-spectral face recognition at a long distance remained difficult to solve. The process of extracting Gauss-SIFT feature was performed on the original image resolution. Thus, their proposed method was not specifically designed for cross-distance face matching. Kang et al. [147] extended the previous work [9] by applying Locally Linear Embedding (LLE) based image restoration method on low-quality images at long distances. Then the restored images were matched against the high-quality gallery images using a HFR matcher. This proposed method can effectively handle the cross-distance matching problem since the quality of restored images could be significantly improved. However, the experimental results were reported on a portion of LDHF-DB\(^1\) and the learning-based image restoration method might not be generalizing well to unseen images.

\(^1\)The training and testing are conducted on the same database based on cross-validation protocol.
Figure 6.7: An NIR probe image acquired at a distance of 150 meter (150m) is matched against a VIS gallery image at a distance of 1 meter (1m). The quality of NIR image is severely degraded by sensor noise, blur and other covariates.

To overcome these limitations, we have developed a HFR matcher capable of matching remote NIR to VIS facial images. The proposed HFR matcher has two distinct advantages: (a) no learning-based image restoration method is required, and (b) an implicit multiresolution matching is automatically integrated. The essence of this method can be divided into two parts. In the first part, Pyramid-SIFT [149] is employed to perform face matching at multiresolutions for face matching between low-resolution and high-resolution face images. The second part is the use of RS-LDA method to learn multiple common discriminant subspaces for face matching between VIS and NIR images. Besides, we also simulate the training dataset with low resolution and blurring effects. The idea is to provide moderate degradations in the training dataset so that the learned matcher can better adapt to the cross-distance face matching in the test set. We do not add noise information here, since the addition of noise actually decreases the performance of the matcher.

The key contributions of our work are listed as follows:

- A comprehensive analysis on different types of image filters that used for cross-spectral face recognition are conducted.
We introduced an effective face descriptor that is used for cross-distance face matching.

We propose a framework that is designed specifically for cross-spectral and cross-distance face matching.

### 6.2.1 Face Descriptors

Prior to the extraction of face descriptors, each face image is first geometrically normalized using an affine transformation based on facial landmarks\(^2\) in order to remove variations due to scale and pose. All normalized face images are cropped and resized to a dimension of 160 × 125. Differing from work of [147] where three different types of imagecroppings (tight cropping, median cropping and loose cropping) were used, we only use median cropping\(^3\) in our experiments in order to remove most hair information from both VIS and NIR images [147]. A median filter with a window size of 7 × 7 is applied to NIR images captured at distances of 60, 100 and 150 meters. This is necessary in order to mitigate the noise impact when NIR images are acquired at long distances. Afterward, a face feature extractor is proceeded as follows: (a) image filtering, and (b) feature extraction.

The process of applying image filters is crucial to the success of cross-spectral face matching [81, 147]. Because VIS and NIR images are acquired under two different spectra, the variability of their appearance is noticeable (see Figure 6.7). Hence, direct matching between VIS and NIR images often resulted in very poor performances [111, 65, 81]. In the work of [65] and [147], three different types of image filters were used: (a) Difference of Gaussian (DoG); (b) Center-Surround Divisive Normalization (CSDN); (c) Gaussian Blur. However, the impact of these filters are not systematically evaluated and the rationales behind these image filters are not well interpreted. Further, their utilization of different image filters targets the feature-level fusion, instead of score-level fusion. In other words, features extracted from individual image filters are concatenated together to form a holistic descriptor. The limitation of this scheme during RS-LDA learning is that patches selected across different feature sets may not be uniformly distributed.

---

\(^2\)Landmarks were manually labeled for NIR face images at distances of 60m and higher.

\(^3\)Leveraging knowledge from all three image cropping will improve the matching performance.
Image Filters

In this work, we use three image filters that are used for heterogeneous face matching. The rationales for selecting these filters will be explained later.

**CSDN**: The first image filter is Center-surround Divisive Normalization (CSDN) filter \([150]\), which has been used as a preprocessing step to extract Biologically Inspired Features (BIF). The CSDN filter divides the value of each pixel by the mean pixel value within the patch:

\[
I'(x,y) = \frac{I(x,y)}{I(x,y) * M(x,y)},
\]

(6.21)

where \(I(x,y)\) is the original image and \(M(x,y)\) is the average filter with size \(s \times s\). Here, \(s = 16\) is used in the implementation.

**GIST**: We utilized a second filter that was originally used in the computation of GIST descriptor \([151]\). Here, the filter is simply denoted as GIST filter. The mathematical formulation of this filter is given by:

\[
I'(x,y) = \frac{I(x,y) * h(x,y)}{\epsilon + \sqrt{[I(x,y) * h(x,y)]^2 * G(x,y)}},
\]

(6.22)

where \(G(x,y)\) is the low-pass Gaussian filter and \(h(x,y) = 1 - G(x,y)\) is the corresponding high-pass filter. Here, \(\epsilon = 0.2\) is used in the implementation. It is noted that GIST filter has not been used before for cross-spectral face matching.

**SQI**: According to the Lambertian model, the image formation process is described as follows \([152]\):

\[
I(x,y) = \rho_w(x,y)n(x,y)s,
\]

(6.23)

where \(\rho_w(x,y)\) is the albedo of the facial surface, \(n\) is the surface normal and \(s\) is the lighting reflection. The self-quotient image, \(Q\), of \(I\) is defined as \([107]\),

\[
Q = \frac{I(x,y)}{\hat{I}(x,y)} = \frac{\rho_w(x,y)n(x,y)s}{G(x,y) * [\rho_w(x,y)n(x,y)s]},
\]

(6.24)

where \(\hat{I}\) is the smoothed version of \(I\) and \(G\) is the Gaussian smoothing kernel.

Among three described image filters (CSDN, GIST and SQI), they share some common properties: low-pass filtering and divisive normalization. CSDN can be considered as a
simplified version of SQI. Both $M(x, y)$ and $G(x, y)$ are low-pass filters. As pointed out in [152], VIS and NIR images can be modeled separately by:

$$I_V(x, y) = \rho_V(x, y)S_V(x, y),$$

and

$$I_N(x, y) = \rho_N(x, y)S_N(x, y),$$

where $S_V(x, y)$ and $S_N(x, y)$ are the large-scale components of the illumination patterns. These large-scale components are assumed to vary smoothly. Within a small region of a face image, $\rho_V(x, y)$ and $\rho_N(x, y)$ are considered to be approximately proportional to each other. Hence, the division between $I_V(x, y)$ and its smoothed version $\hat{I}_V(x, y)$ is approximately equal to the division of $I_N(x, y)$ and $\hat{I}_N(x, y)$.

The results of applying these image filters are shown in Figure 6.8. Visually, GIST-based image filtering can maximally restore the degraded face image. However, this does not always yield better classification performances. We have also studied other image filters, such as DoG\(^4\) and Gaussian Blur in [147]. Since we already consider the Gaussian blur during the feature representation via Pyramid-SIFT, it is not utilized as an independent image filter. The difference among these image filters is how to approximate the illumination pattern and remove it.

**Pyramid-SIFT Representation**

After applying image filters to pre-process face images, we adopt Pyramid-SIFT feature extractor [149] to capture discriminative features at multiresolutions for both VIS and NIR images. The underlying idea is to compute feature descriptors densely from a fixed set of points, rather than these obtained from potentially unreliable interest point detectors [9, 149]. In particular, we sample patches with a size of $24 \times 24$ pixels and a stride of 12 pixels. Each sampled patch is divided into $4 \times 4$ subgrids. A magnitude-weighted gradient orientation histogram with a bin number of 8 is calculated, resulting in a 128 ($4 \times 4 \times 8$) dimensional feature vector. The spatial information is also considered here. Thus, the dimension of the

\(^4\)Fusion with DoG image filter does not lead to improved matching performance.
Figure 6.8: Exemplar images of the same subject from LDHF database after the process of image filtering. Top row shows the results after applying image filters to a visible image at 1m distance; Bottom row shows the results after applying image filters to a near-infrared image at 150m distance.

Figure 6.9: An illustration of Pyramid-SIFT feature extraction at five different image resolutions. The blurring effect increases with the downsampling factor. The number of patches extracted from each image resolution is 108, 48, 24, 12 and 4. For visualization purpose, the image is upscaled to the original resolution.

The final feature vector is 130. This process is repeated for five different image resolutions, with a downsampling factor of $\sqrt{2}$. The feature extraction process is illustrated in Figure 6.9.

Overall, the total number of patches extracted from an image is 196. In the work of [9], Gauss-SIFT (GSIFT) was used, which extracted a total number of 154 patches. Differing from Gauss-SIFT, we refer to this method as Pyramid-SIFT (PSIFT). The major difference between GSIFT and PSIFT is that GSIFT does not take into consideration the “multiresolution” information of face images at varying camera distances.

### 6.2.2 Random Subspace Linear Discriminant Analysis (RS-LDA)

In this section, we briefly describe the use of RS-LDA method for matching NIR images to VIS images. For each of the three image filters (CSDN, GIST, and SQI), the following
procedure is adopted. The resultant matching scores are then fused from three image filters to arrive at a final decision.

**Patch Extraction:** To extract discriminative features, an image is convolved with a Gaussian filter and then downscaled to different resolutions, resulting in a tessellation of $P$ patches. Therefore, a set of $P$ feature vectors are extracted from a given image. Let the set of feature vectors extracted from a VIS image ($I_{i,j}^V$) be denoted as $X_{i,j}^V = f(I_{i,j}^V)$. The $p$-th feature vector from $X_{i,j}^V$ is denoted as $X_{i,j}^V(p)$, where $X_{i,j}^V(p) \in \mathbb{R}^{130}$ for SIFT features. Similarly, $X_{i,j}^N = f(I_{i,j}^N)$ is used to denote the set of feature vectors from an NIR image ($I_{i,j}^N$).

**Patch Sampling:** For creating the $k$-th subspace, $k = \{1, 2, \ldots, K\}$, we sample $\alpha$ number of patches without replacement from $X_{i,j}^V$ and $X_{i,j}^N$, where $i = \{1, \ldots, c\}$, $j = \{1, \ldots, n_i\}$ for VIS and $j = \{1, \ldots, m_i\}$ for NIR, and $\alpha \in \{1, \ldots, P\}$, to obtain $x_{i,j}^V \in \mathbb{R}^{D \times 1}$ and $x_{i,j}^N \in \mathbb{R}^{D \times 1}$. Here, $x_{i,j}^V$ and $x_{i,j}^N$ are obtained by concatenating all the $\alpha$ feature vectors into a single vector representation, where $D = \alpha \times d$ and $d$ is the feature dimension for each patch. Because the dimensionality of these vectors $x_{i,j}^V$ and $x_{i,j}^N$ is usually very high, they are first reduced via the Principal Component Analysis (PCA) [65]. The projected feature vectors are denoted as $y_{i,j}^V$ and $y_{i,j}^N$, respectively. The associated PCA subspace in the $k$-th iteration is referred to as $W_{E}^{(k)}$.

**Discriminant Subspace:** To learn the common discriminant subspace, we consider both VIS and NIR images for within-class and between-class constructions. The mean class vector for the $i$-th subject when constructing the $k$-th subspace for a feature descriptor is calculated using both VIS ($y_{i,j}^V$) and NIR projected vectors ($y_{i,j}^N$): $\mu_{i}^{(k)} = \frac{1}{n_i + m_i} \left( \sum_{j=1}^{n_i} y_{i,j}^V + \sum_{j=1}^{m_i} y_{i,j}^N \right)$. Then the between-class scatter matrix can be computed by:

$$S_{B}^{(k)} = \sum_{i=1}^{c} (\mu_{i}^{(k)} - \mu_{i}^{(k)}) (\mu_{i}^{(k)} - \mu_{i}^{(k)})^T,$$

where $\mu_{i}^{(k)} = \frac{1}{c} \sum_{i=1}^{c} \mu_{i}^{(k)}$. The within-class scatter matrix can be computed by:

$$S_{W}^{(k)} = \sum_{i=1}^{c} \sum_{j=1}^{n_i} (y_{i,j}^V - \mu_{i}^{(k)}) (y_{i,j}^V - \mu_{i}^{(k)})^T + \sum_{i=1}^{c} \sum_{j=1}^{m_i} (y_{i,j}^N - \mu_{i}^{(k)}) (y_{i,j}^N - \mu_{i}^{(k)})^T.$$

(6.27)
The projection matrix $V^{(k)}$ for discriminant subspace can be obtained by solving the generalized eigenvalue problem of $S_B^{(k)}V^{(k)} = \lambda^{(k)}S_W^{(k)}V^{(k)}$.

**Classification:** At the end of each subspace iteration $k$, both VIS and NIR samples are first projected onto $W_E^{(k)}$ and then the discriminant subspace $V^{(k)}$. Finally, the VIS samples are used in the gallery to form as $Y^V$ and the NIR sample $y_{i,j}^N$ is used as a probe. The coefficient of projection of an NIR sample can be obtained using SRC solution [143]:

$$\rho^{(k)} = \arg \min_\rho \|Y^V \rho - y_{i,j}^N\|_2^2 + \lambda \|\rho\|_1,$$  

(6.29)

where $\lambda$ is the regularization term. The coefficient vectors pertaining to all subspaces are summed together: $\rho = \sum_1^K \rho^{(k)}$. The match score between the $r$-th entry in the gallery and the probe image corresponds to the $r$-th element in $\rho$.

### 6.2.3 Whitening Transformation

During the computation of $W_E^{(k)}$, both VIS and NIR samples were used in this process and they were treated as samples generated from same modality (or spectrum). However, the difference between VIS and NIR spectra is significant. Thus, the label information from different spectra can be utilized to improve the subspace learning. In the work of [153, 154, 155], whitening transformation is used for PCA in order to build more robust subspaces. First, all instances of $x_{i,j}^V$ and $x_{i,j}^N$ are combined as $X^V$ and $X^N$, respectively. Then, the cross-spectral mean for the projected vectors are calculated as:

$$C^{(k)} = W_E^{(k)} * (X^V + X^N)/2.$$  

(6.30)

This is used to center the VIS and NIR training samples,

$$\tilde{X}^V = W_E^{(k)} * X^V - C^{(k)},$$  

(6.31)

$$\tilde{X}^N = W_E^{(k)} * X^N - C^{(k)}.$$  

(6.32)

To reduce the intra-personal variation between the VIS and NIR samples, a whitening transform is applied. This is preceded by recombination of VIS and NIR samples as the entire matrix $\tilde{X} = [\tilde{X}^V \tilde{X}^N]$ and then perform the PCA analysis to obtain the eigenvectors $\tilde{W}_{E1}$.
The whitening transform process is applied:

\[ \tilde{W}_{E_1} = (\Lambda^{-\frac{1}{2}} \tilde{W}_{E_1}^T)^T. \] (6.33)

Here, \( \Lambda \) is a diagonal matrix whose entries are the corresponding eigenvalues of PCA eigenvectors of \( \tilde{W}_{E_1} \). After that, another projection matrix \( \tilde{W}_{E_2} \) is computed by performing PCA on \( \tilde{W}_{E_1} C^{(k)} \). The final PCA subspace is \( W_F^{(k)} = W_E^{(k)} \tilde{W}_{E_1} \tilde{W}_{E_2} \). Therefore, we can replace \( W_E^{(k)} \) in Section 6.2.2 with \( W_F^{(k)} \). The following common discriminant subspace learning remains the same. This whitening transformation requires the number of \( X^V \) and \( X^N \) samples to be the same.

### 6.2.4 Database

The following experiments were designed with the primary goal of exploring the effectiveness of the proposed scheme in matching remote NIR to VIS face images. We use the following parameters in our work for RS-LDA: \( \alpha = 60 \) and \( K = 20 \).

To validate the effectiveness of the proposed method, we test the proposed face matcher on LDHF-DB [9]. LDHF-DB contains both visible and near-infrared face images at distances of 1m indoor, 60m, 100m and 150m outdoor. For each scenario, it consists of 100 subjects, where each subject has one sample image. Some of the exemplar images at different distances are shown in Figure 6.10. The quality of the NIR images starts to degrade with increase of camera distance. Such degradations include noise, blur, low resolution and weather artifacts that are typically observed for long range facial image acquisition [156]. In addition to the aforementioned dataset, we utilized HFB dataset [148] for the purpose of training the face matcher. It contains 200 subjects, with 2,095 VIS images and 2,980 NIR images. For the whitening transformation, 1,600 VIS and 1,600 NIR images were used for the training.

### 6.2.5 Intra-spectral and Cross-distance Face Matching

To demonstrate the effectiveness of PSIFT face descriptor for cross-distance face matching, we evaluate it on the intra-spectral and cross-distance face matching experiment. This matching scenario typically occurs in daytime surveillance where cameras are used to capture remote face images. Here, we use face matching between visible images (VIS-VIS) as
an example. Hence, 60m VIS, 100m VIS and 150m VIS are used as probes, while 1m VIS face images are used in the gallery. Correspondingly, they are denoted as 1mVIS-60mVIS, 1mVIS-100mVIS and 1mVIS-150mVIS. Because there is only one image per subject, we cannot consider the 1mVIS-1mVIS matching.

As seen from Figure 6.11, PSIFT descriptor with CSDN filter achieves Rank-1 accuracies of 88.0%, 80.0% and 62.0% for 1mVIS-60mVIS, 1mVIS-100mVIS and 1mVIS-150mVIS, respectively. GIST filter obtains the best Rank-1 accuracy (77.0%) for 1mVIS-150mVIS. When no image filters are applied, PSIFT obtains 75.0%, 69.0%, and 67.0%, respectively. Because it is an intra-spectral matching, applying image filters do not necessary guarantee improved performance. However, applying GIST filter dose improve performances on all three matching scenarios. In a summary, the utilized PSIFT descriptor has achieved good performances on cross-distance face matching.

6.2.6 Cross-spectral and Cross-distance Face Matching

In this experiment, cross-spectral and cross-distance face matching is considered. This matching scenario occurs in nighttime surveillance where NIR cameras are used to capture remote face images and matched against the VIS mugshots stored in the database. Therefore, 1m VIS images are used in the gallery and NIR images at a long distance (60m, 100m or
Evaluation of Image Filters

Similar to Section 6.2.5, we first attempt to evaluate the performance of the PSIFT descriptor without RS-LDA subspace learning. As mentioned in the scenario of intra-spectral and cross-distance face matching, the effectiveness of these image filters are not convincing. Here, we proceed to demonstrate the roles of these image filters in cross-spectral face matching.

As can be seen from Figure 6.12, direct matching between VIS and NIR images results in very poor performance. It obtains Rank-1 accuracies of 56.0%, 36.0%, 20.0% and 8.0% for 1mVIS-1mNIR, 1mVIS-60mNIR, 1mVIS-100mNIR, and 1mVIS-150mNIR, respectively. After applying image filters, the Rank-1 performances dramatically improve. For instance, CSDN filter obtains 93.0%, 66.0%, 47.0% and 21.0% for individual matching scenarios. This confirms our observation that image filters are crucial to the success of cross-spectral face matching.

Though good performances have been achieved for 1mVIS-1mNIR face matching, chal-
Figure 6.12: Rank-1 performance evaluation of PSIFT descriptor with different image filters on LDHF-DB. Cross-spectral and cross-distance face matching is considered here.

Table 6.3: Evaluation of CSDN image filter on cross-spectral and cross-distance face matching. GAR (%) performances at both FARs of 0.1% and 1% are reported, along with Rank-1 accuracy (%).

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>0.1% FAR</th>
<th>1% FAR</th>
<th>Rank-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1mVIS-1mNIR</td>
<td>98.26</td>
<td>100.0</td>
<td>97.0</td>
</tr>
<tr>
<td>1mVIS-60mNIR</td>
<td>67.91</td>
<td>89.0</td>
<td>85.0</td>
</tr>
<tr>
<td>1mVIS-100mNIR</td>
<td>40.48</td>
<td>64.50</td>
<td>60.0</td>
</tr>
<tr>
<td>1mVIS-150mNIR</td>
<td>11.0</td>
<td>30.0</td>
<td>24.0</td>
</tr>
</tbody>
</table>

Challenges remain when the distance of NIR probe images increases. Next, we demonstrate that by applying RS-LDA method, the performance can be further improved. We report Genuine Accept Rate (GAR) at 0.1% False Accept Rate (FAR) and 1% False Accept Rate (FAR), in addition to Rank-1 accuracy.

As seen from Table 6.3, CSDN obtains GARs of 89.0%, 64.50% and 30.0% at 1% FAR for distances of 60m, 100m and 150m, respectively. The reported Rank-1 accuracies are 85.0%, 60.0% and 24.0%.

As seen from Table 6.4, GIST obtains GARs of 89.05%, 75.0% and 38.62% at 1% FAR for distances of 60m, 100m and 150m, respectively. Compared to CSDN filter, the performance is further improved. That also indicates that the proposed GIST filter is more suitable in handling cross-spectral and cross-distance face matching. In other words, it is less sensitive
Table 6.4: Evaluation of GIST image filter on cross-spectral and cross-distance face matching. GAR (%) performances at both FARs of 0.1% and 1% are reported, along with Rank-1 accuracy (%).

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>0.1% FAR (%)</th>
<th>1% FAR (%)</th>
<th>Rank-1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1mVIS-1mNIR</td>
<td>95.0</td>
<td>100.0</td>
<td>97.0</td>
</tr>
<tr>
<td>1mVIS-60mNIR</td>
<td>79.01</td>
<td>89.05</td>
<td>85.0</td>
</tr>
<tr>
<td>1mVIS-100mNIR</td>
<td>38.96</td>
<td>75.0</td>
<td>67.0</td>
</tr>
<tr>
<td>1mVIS-150mNIR</td>
<td>6.972</td>
<td>38.62</td>
<td>34.0</td>
</tr>
</tbody>
</table>

Table 6.5: Evaluation of SQI image filter on cross-spectral and cross-distance face matching. GAR (%) performances at both FARs of 0.1% and 1% are reported, along with Rank-1 accuracy (%).

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>0.1% FAR (%)</th>
<th>1% FAR (%)</th>
<th>Rank-1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1mVIS-1mNIR</td>
<td>93.0</td>
<td>98.75</td>
<td>95.0</td>
</tr>
<tr>
<td>1mVIS-60mNIR</td>
<td>54.72</td>
<td>80.22</td>
<td>77.0</td>
</tr>
<tr>
<td>1mVIS-100mNIR</td>
<td>30.96</td>
<td>67.0</td>
<td>58.0</td>
</tr>
<tr>
<td>1mVIS-150mNIR</td>
<td>16.0</td>
<td>31.57</td>
<td>26.0</td>
</tr>
</tbody>
</table>

to image degradations.

As seen from Table 6.5, SQI obtains GARs of 80.22%, 67.0% and 31.57% at 1% FAR for distances of 60m, 100m and 150m, respectively. Compared to CSDN and GIST image filters, SQI has slightly worse performance.

**Proposed Method**

Each image filter has demonstrated its own strength in cross-spectral and cross-distance face matching (see Section 6.2.6). Considering that image filters can provide complementary information in classification, we perform the score-level fusion from all image filters. As can be seen from Table 6.6, we achieve Rank-1 accuracies of 87.0%, 70.0% and 43.0% at distances of 60m, 100m, 150m, respectively. The reported GARs at 0.1% FAR and 1% FAR are also significantly better than individual image filters. This clearly validates the benefits of fusing outputs from all image filters. Meanwhile, we also evaluate the performance with whitening transformation in Table 6.7. Not surprisingly, after applying whitening transformation, we
Table 6.6: CSDN+GIST+SQI: GAR performances at both FARs of 0.1% and 1% are reported, along with Rank-1 accuracy.

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>0.1% FAR</th>
<th>1% FAR</th>
<th>Rank-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1mVIS-1mNIR</td>
<td>99.0</td>
<td>100.0</td>
<td>99.0</td>
</tr>
<tr>
<td>1mVIS-60mNIR</td>
<td>76.81</td>
<td>91.0</td>
<td>87.0</td>
</tr>
<tr>
<td>1mVIS-100mNIR</td>
<td>52.0</td>
<td>77.66</td>
<td>70.0</td>
</tr>
<tr>
<td>1mVIS-150mNIR</td>
<td>21.0</td>
<td>44.0</td>
<td>43.0</td>
</tr>
</tbody>
</table>

Table 6.7: CSDN+GIST+SQI (Whitening): GAR performances at both FARs of 0.1% and 1% are reported, along with Rank-1 accuracy.

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>0.1% FAR</th>
<th>1% FAR</th>
<th>Rank-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1mVIS-1mNIR</td>
<td>100.0</td>
<td>100.0</td>
<td>99.0</td>
</tr>
<tr>
<td>1mVIS-60mNIR</td>
<td>83.0</td>
<td>94.0</td>
<td>88.0</td>
</tr>
<tr>
<td>1mVIS-100mNIR</td>
<td>52.9</td>
<td>80.0</td>
<td>68.0</td>
</tr>
<tr>
<td>1mVIS-150mNIR</td>
<td>20.33</td>
<td>36.0</td>
<td>37.0</td>
</tr>
</tbody>
</table>

have improved the GAR performances at 60m and 100m outdoors. The performance decreases slightly for 150m outdoor scenario. Since the whitening transformation is performed on the training dataset HFB (close to 1mVIS-1mNIR), it may not extend well to reduce the cross-spectral difference in the test set for images captured at longer distances (e.g., 150m).

Comparison with State-of-the-art

The proposed method is also evaluated against existing methods that are published on the LDHF-DB (see Table 6.8). The proposed method (without whitening) exceeds the previous work (GSIFT) by 10.0%, 16.66%, 24.0% at 1% FAR on 60m, 100m, and 150m, respectively. The whitening transformation improves the matching performance for 60m and 100m. The evaluation results from [147] are generated by performing training and testing on the same LDHF-DB. Therefore, their test results are reported only on a small portion of the dataset. Though they achieve much better performance, the results cannot be directly compared with

---

5 The results are directly cited from authors’ paper.
Figure 6.13: Reported ROC curve for the cross-spectral and cross-distance face recognition for the proposed method (without whitening).

Table 6.8: Comparison against the HFR matchers (GSIFT and GSIFT+COTS) in [9]. The proposed method with or without whitening transformation is reported in the last column. GAR (%) performances at 1% FAR are reported.

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>GSIFT</th>
<th>GSIFT+COTS</th>
<th>Proposed/Whitening</th>
</tr>
</thead>
<tbody>
<tr>
<td>1mVIS-1mNIR</td>
<td>NULL</td>
<td>NULL</td>
<td>100.0/100.0</td>
</tr>
<tr>
<td>1mVIS-60mNIR</td>
<td>81.0</td>
<td>89.0</td>
<td>91.0/94.0</td>
</tr>
<tr>
<td>1mVIS-100mNIR</td>
<td>61.0</td>
<td>63.0</td>
<td>77.66/80.0</td>
</tr>
<tr>
<td>1mVIS-150mNIR</td>
<td>20.0</td>
<td>18.0</td>
<td>44.0/36.0</td>
</tr>
</tbody>
</table>

6.3 Matching Thermal to Visible

In previous section, we have proposed an effective subspace learning scheme for face recognition between remote near-infrared and visible face images. Another similar, but more challenging problem is the face matching between thermal and visible face images. The use of thermal infrared (THM) images provides a feasible solution for illumination invariant face recognition due to the fact that infrared spectrum is less sensitive to changes in illumination [60]. Thermal emissions from the face are an intrinsic property that is relatively less affected by ambient illumination [60]. However, images in legacy databases are often captured in the visible spectrum (VIS). Therefore, matching THM face images to
VIS face images, often referred to as Heterogeneous Face Recognition (HFR), is of particular importance in designing nighttime face recognition systems [157, 65].

To date, there is limited scientific literature on addressing the challenge of matching THM to VIS face images (see Figure 6.14). Li et al. [158] proposed a learning-based framework to synthesize VIS face images from THM face images. The synthesized VIS face images were then matched against the original VIS face images. They reported a Rank-1 accuracy of 50.06% on the Equinox database. Choi et al. [157] utilized Partial Least Squares Discriminant Analysis (PLS-DA) to learn the correlation between VIS and THM face images. In their work, image filters based on Self Quotient Image (SQI) and Difference of Gaussian (DoG) were used, along with four different texture descriptors. Their experiments on the UND dataset yielded a Rank-1 accuracy of 49.9%. Klare and Jain [65] proposed a HFR matching framework called Prototype Random Subspace (P-RS) that utilized three different image filters: DoG, Center-Surround Divisive Normalization (CSDN), and Gaussian. The reported Rank-1 accuracy was approximately 50.0% on the PCSO dataset.

The performance numbers above suggest that there is plenty of scope for improvement. Further, in the approaches cited above [158, 157, 65] the training and test images were taken from the same database. i.e., the cross-database matching scenario was not explored. This paper concerns the development of a novel HFR framework for matching THM face images to VIS face images (see Figure 6.15).
Figure 6.15: The proposed framework for matching thermal to visible face images. A cascaded subspace learning process (whitening transformation, factor analysis and common discriminant analysis) involving both VIS and THM images is used during the training phase. The Gallery and Probe samples are then projected onto the learned subspaces during the testing phase.
The underlying scheme is an ensemble learning approach to generate multiple common random subspaces for VIS and THM images. Each subspace is based on a set of randomly selected patches from the input image (VIS and THM). First, each image in the training set is convolved with an image filter. The filtered image is tessellated into patches. Each patch is represented by a feature descriptor and the descriptors from the subset of patches are combined into a single vector. The training vectors are then used to derive a whitening transform, and a factor analysis model is used to extract spectral-invariant features from the transformed vector. Finally, Linear Discriminant Analysis (LDA) is used to derive a common discriminant subspace for THM and VIS images. This proposed matching framework is referred to as RSLDA+Whitening+HFA. RSLDA refers to the matching framework without the whitening transformation and factor analysis layers; RSLDA+Whitening refers to the framework without the factor analysis layer.

The proposed approach is motivated by the following observations: (1) A common discriminant subspace between VIS and THM cannot be learned accurately by using a single type of subspace transformation due to the large appearance difference; (2) The prior knowledge about which face region is significant to the learning is unknown and, therefore, random sampling of patches is essential; (3) There exists a spectral-invariant identity factor between VIS and THM that has to be carefully elicited from the two participating spectral bands.

6.3.1 Proposed Matching Framework

The main component of the proposed framework is cascaded subspace learning that is used to extract identity features. The cascaded subspace learning scheme has three different layers- whitening transformation, factor analysis, common discriminant analysis - with each layer serving a different purpose. This section briefly introduces the whitening transformation layer. The discussion on the factor analysis model layer will be elaborated in the subsequent section.
To reduce some of the cross-spectral appearance difference, three image filters are used: CSDN [65], SQI [157] and GIST [151]. GIST has never been explored in cross-spectral matching. Examples of VIS and THM images after the application of the image filters are shown in Figure 6.16. This clearly shows the potential of reducing the appearance difference between VIS and THM. While this photometric adjustment is useful, it does not necessarily help in extracting identity-specific features.

After the application of image filters, the PSIFT descriptor [149] is utilized to describe the face in both VIS and THM spectra. PSIFT is a variant of the SIFT descriptor [149] and it extracts patches centered on densely sampled points at different image resolutions. To compute the descriptor, both magnitude and orientation of the gradients are calculated for each patch. Then, each patch is divided into $4 \times 4$ subgrids, where an 8-bin magnitude-weighted gradient orientation histogram is calculated in each subgrid. Finally, histograms are concatenated to form a 128-dimensional feature vector ($4 \times 4 \times 8$). The dimension of PSIFT feature vector per patch is 130, which includes spatial coordinate information. In addition to PSIFT, we utilized a second descriptor named Histograms of Principal Oriented Gradients (HPOG) [159] which is derived based on Histograms of Oriented Gradients (HOG) [160]. Rather than computing pixel-wise gradients as in HOG that might be sensitive to local appearance changes resulting from expression, noise and low-resolution, HPOG computes...
the principal gradient of a neighboring region centered around a pixel and uses that gradient value when computing the histogram. The principal gradient is the largest eigenvector of the covariance matrix of the pixel gradients within the region. The dimension of HPOG feature vector per patch is 108.

As depicted in Figure 6.15, multiple subspaces (K) are used to generate the ensemble of classifiers corresponding to each image filter. Each subspace is constructed based on the random sampling of image patches. For creating the \( k \)-th subspace, \( k = \{1, 2, \ldots, K\} \), \( \alpha \) number of patches are sampled (without replacement) pertaining to a specific descriptor (PSIFT or HPOG) and concatenated to form a feature vector. These feature vectors are computed from all images in VIS and THM training sets, which are depicted as matrices \( X^V \) and \( X^T \).

**Whitening Transformation**

Both VIS and THM training samples are used in this process. First, the overall mean of the computed feature vectors is calculated as:

\[
C^{(k)} = W_E^{(k)} \cdot (X^V + X^T)/2, \quad (6.34)
\]

where \( W_E^{(k)} \) are the eigenvectors computed from the covariance matrix based on \( X^V \) and \( X^T \) in the \( k \)-th subspace. This is used to center the VIS and THM training samples,

\[
\tilde{X}^V = W_E^{(k)} \cdot X^V - C^{(k)}, \quad (6.35)
\]

\[
\tilde{X}^T = W_E^{(k)} \cdot X^T - C^{(k)}. \quad (6.36)
\]

To reduce the variation between VIS and THM samples, a whitening transform is applied. This is preceded by recombination of VIS and THM samples as the entire matrix \( \tilde{X} = [\tilde{X}^V \tilde{X}^T] \) and then perform the PCA analysis to obtain the eigenvectors \( \tilde{W}_{E1} \). The whitening transform process is applied via:

\[
\tilde{W}_{E1} = (\Lambda^{-\frac{1}{2}}(\tilde{W}_{E1})')'. \quad (6.37)
\]

Here, \( \Lambda \) is a diagonal matrix whose entries are the corresponding eigenvalues of PCA eigenvectors of \( \tilde{W}_{E1} \). After that, another projection matrix \( \tilde{W}_{E2} \) is computed by performing
PCA on $\hat{W}_E C^{(k)}$. The final PCA subspace is $W_F^{(k)} = W_E^{(k)} \hat{W}_E \hat{W}_E$. This whitening transformation requires the number of $X^V$ and $X^T$ samples to be the same.

### 6.3.2 Factor Analysis Model

After whitening transformation layer, the ensuing layer uses a factor analysis model to learn another type of subspace transformation.

#### Motivation

In this section, we describe the characteristics of Factor Analysis Model (FAM) [161, 154]. FAM is also referred to as Latent Variable Model [162]. A factor analysis model seeks to decompose an observed vector $x \in \mathbb{R}^d$ into a corresponding Gaussian latent variable $z \in \mathbb{R}^m$ plus an additive Gaussian noise variable:

$$x = Wz + \mu + \epsilon,$$

(6.38)

where $W \in \mathbb{R}^{d \times m}$ is the latent subspace, $\mu$ is the mean vector, and $\epsilon$ is the Gaussian noise variable with diagonal covariance matrix $\Psi \in \mathbb{R}^{d \times d}$. The factor analysis model is closely related to Probabilistic PCA [162]. The parameters of $\mu$, $W$, and $\Psi$ can be determined by Maximum Likelihood Estimation (MLE). Because there exists no closed-form maximum likelihood solution for $W$, Expectation Maximization (EM) algorithm is used to seek the solution in an iterative way [162].

Using FAM for face recognition has been studied before in [161, 154]. It is well known that the identity of an individual is impacted by intra-class variations such as pose, illumination and expression [59]. Given an observed vector $x$, we seek to recover the latent variable $z$ that truly represents the identity of an individual, regardless of associated variations. Hence, a face image can be factorized into two main components: an identity-specific component $Wz$ and the other component $\mu + \epsilon$ that reflects the intra-class variations.

#### Hidden Factor Analysis

In the work of [154], Hidden Factor Analysis (HFA) model was proposed to extract person-specific features that were stable across age variations. Their model decomposed an
observed vector into two latent factors (or variables): an age-invariant identity factor and an age factor that was related to the aging process. Their experiments on Morph and FGNET datasets verified the effectiveness of HFA in age-invariant face recognition.

However, there is no prior work on studying HFA model in cross-spectral face recognition. Similar to the work of [154], the main idea is that a face image of an individual can be regarded as the combination of two components: an identity component that is invariant across spectra, and another component that describes the spectral variation. In practical scenarios, there always exist noise factors. Therefore, a face image is decomposed into three components: (a) an identity component; (b) a spectrum component; and (c) a noise component. For simplicity, a linear generative model is used, which means that a face image is a linear combination of these three components. This linear model can be described as [154]:

$$t = \beta + Ux + Vy + \epsilon,$$  \hspace{1cm} (6.39)

where $t \in \mathbb{R}^d$ is the observed vector that represents the feature vector extracted from a face image, $\beta \in \mathbb{R}^d$ is the mean vector, $x \in \mathbb{R}^p$ is the latent identity factor with a prior distribution of $N(0, I)$, $y \in \mathbb{R}^q$ is the latent spectrum factor with a prior distribution of $N(0, I)$, $\epsilon \in \mathbb{R}^d$ is a noise vector with a prior distribution of $N(0, \sigma^2 I)$, $U \in \mathbb{R}^{d \times p}$ is a matrix whose columns represent the bases for the identity subspace, and $V \in \mathbb{R}^{d \times q}$ whose columns represent the bases for the spectrum subspace. In summary, any observed face feature vector contains three components: the identity component $Ux$, the spectrum component $Vy$ and the other component $\beta + \epsilon$.

The HFA model has a set of parameters $\Theta = \{\beta, U, V, \sigma^2\}$ to be estimated. These parameters can be learned from the training data using the MLE algorithm [162]. The objective function is defined as:

$$L_c = \sum_{i,k} \log p_{\Theta}(t^k_i, x_i, y_k).$$  \hspace{1cm} (6.40)

Here, $t^k_i$ denotes the observed feature vector for $i$-th subject in the $k$-th spectrum. $x_i$ is the corresponding identity factor and $y_k$ is the corresponding spectrum factor. In Eqn. (6.40), we have two unknown latent variables $x_i$ and $y_k$ in addition to $\Theta$. This can
be solved by updating the model parameter $\Theta$ while fixing the latent variables and vice versa. For instance, given the initial model parameter $\Theta_0$, the posterior distribution of latent variables of $x_i$ and $y_k$ can be estimated by $p_{\Theta_0}(x_i, y_k|T)$. Here, $T$ is the training data from both VIS and THM. Next, $\Theta$ can be updated by maximizing the following objective function:

$$\sum_{i,k} \int p_{\Theta_0}(x_i, y_k|T) \log p_{\Theta}(t_k, x_i, y_k) dx_i y_k. \quad (6.41)$$

More details of the derivations can be found in [154]. Once we obtain the model parameters, the identity factor can be computed as:

$$x = UU^T \Sigma^{-1}(t - \beta), \quad (6.42)$$

where $\Sigma = \sigma^2 I + UU^T + VV^T$.

**Proposed Method**

In our work, HFA model is performed on the sampled patches, instead of partitioned data used in [154]. For $k$-th subspace, a fixed number of patches are randomly sampled from the entire set of patches extracted by PSIFT or HPOG. The randomly sampled patches are concatenated to form an original feature vector. The entire matching framework is detailed as follows.

**Training Phase:** The training process takes both VIS and THM feature vectors as inputs.

1. Whitening Transformation: it aims to reduce the cross-spectral difference by performing whitening transformation on sampled patches from VIS and THM during the PCA subspace learning (see Section 6.3.1). The outputs of the whitening transformation are subspace $W_F$ and the associated mean vector $M_F$.

2. Hidden Factor Analysis: HFA aims to extract identity factor that is invariant to spectral variation. The outputs of HFA are $\beta, U, V, \Sigma$.

3. Linear Discriminant Analysis: LDA learns a common discriminant subspace from VIS and THM feature vectors (obtained from HFA) by taking into consideration the label
information from VIS and THM samples. The output of LDA is the learned common subspace $W_L$.

**Testing Phase:** Let $x^V$ and $x^T$ denote the original extracted feature vectors for VIS and THM spectral images, respectively. To extract final projected feature vectors, the following procedure is adopted.

1. $x^V$ and $x^T$ are first projected onto the whitening transformation subspace:

   \[ y^V = (W_F)'(x^V - M_F), \]  
   \[ y^T = (W_F)'(x^T - M_F). \]  

2. To obtain the identity factors, Eqn. (6.42) is used:

   \[ z^V = UU'\Sigma^{-1}(y^V - \beta), \]  
   \[ z^T = UU'\Sigma^{-1}(y^T - \beta). \]  

3. $z^V$ and $z^T$ are finally projected to the common discriminant subspace:

   \[ f^V = (W_L)'z^V, \]  
   \[ f^T = (W_L)'z^T. \]  

The projected feature vectors $f^V$ and $f^T$ from each subspace $k$ are concatenated to form the final feature vector. NN classifier is used to compute the distance matching score. The matching scores are then fused for all three image filters. In subsequent experiments, the number of dimensions for identity and spectrum factors is 100 and 3, respectively.

6.3.3 **Baseline Algorithms**

The accuracy of the proposed cascaded subspace learning is compared against several heterogeneous face matchers. They are used as baselines in our analysis.
Partial Least Squares (PLS): PLS [163, 164] maps input vectors (regressors) and corresponding output vectors (responses) into a common feature space such that the covariance between the projected input and output vectors is maximized. Let $X \in \mathbb{R}^{m \times N}$ and $Y \in \mathbb{R}^{n \times N}$ denote input and out vectors, respectively. $N$ is the total number of samples. $m$ and $n$ are the dimensions for input and output vectors, respectively. Hence, PLS can be formulated as [163],

$$X = PT + E$$
$$Y = QU + F,$$

where $T \in \mathbb{R}^{p \times N}$ and $U \in \mathbb{R}^{p \times N}$ are factor matrices, $P \in \mathbb{R}^{m \times p}$ and $Q \in \mathbb{R}^{n \times p}$ are loading matrices, and $E \in \mathbb{R}^{m \times N}$ and $F \in \mathbb{R}^{n \times N}$ are residuals. PLS algorithm seeks weight vectors $w$ and $c$ that can maximize the variations preserved in $X$ and $Y$:

$$\max_{w, c} \left[ \text{cov}(Xw, Yc) \right]^2,$$

subject to the constraints: $\|w\| = 1$ and $\|c\| = 1$.

Canonical Correlation Analysis (CCA): CCA [163] aims to search the best correlation between two sets of corresponding feature vectors, which maximizes the following objective function:

$$\max_{w, c} \left[ \text{corr}(Xw, Yc) \right]^2.$$

PLS and CCA methods have been widely used in face matching between near-infrared and visible face images [111, 135, 157]. To ensure a fair comparison between the proposed matcher and PCA/CCA methods, we replace the cascaded subspace learning scheme with PLS/CCA methods. In other words, PLS and CCA methods are embedded within the random subspace. Here, PCA is applied to reduce the feature dimensions. We use RS-PLS and RS-CCA to denote the modified algorithms. The final similarity score is the fusion of matching scores from three different image filters.

The following experiments are conducted with the primary purpose of evaluating the proposed matching framework in addressing face matching between VIS and THM. Both single-database (Section 6.3.4) and cross-database (Section 6.3.6) tests are carried out. The number of subspaces $K$ for each image filter is 20.
Figure 6.17: Samples of face images in visible (top) and thermal (bottom) spectra. Images are from PCSO database and not co-registered.

6.3.4 Evaluation on PCSO dataset

To evaluate the performances of different face matchers, we test them on a dataset collected by Pinellas County Sheriff’s Office (PCSO). The PCSO database contains face images of 1003 subjects, with each subject including two visible images and one thermal image. The original image size is $480 \times 640$. The size of each image after alignment and cropping is $160 \times 125$. Similar to the experimental protocol stated in [65], first 667 subjects were used to train the face matcher and the remaining subjects were used for the testing. There is no overlap between training and testing subjects. Some of the example images are shown in Figure 6.17.

Face Descriptors

As seen from Table 6.9, RSLDA with PSIFT obtains GARs of 13.81% and 36.38% at 0.1% FAR and 1% FAR, respectively. The associated Rank-1 accuracy is 47.92%. Further improvement in identification performance is observed when whitening transformation is applied before LDA. Here, RSLDA+Whitening obtains GARs of 20.57% and 45.10% at 0.1%FAR and 1% FAR, respectively. The corresponding Rank-1 accuracy is 55.36%. Moreover, with the addition of HFA layer, the performance is again significantly improved. RSLDA+Whitening+HFA obtains GARs of 42.42% and 70.13% at 0.1%FAR and 1% FAR, respectively. The obtained Rank-1 accuracy is 71.43%. Similar performances have been observed for HPOG in Table 6.10. Compared with PSIFT descriptor, HPOG obtains slightly
Table 6.9: Evaluation of different face matchers on PCSO thermal dataset with PSIFT descriptor. GAR (%) performances at both 0.1% FAR and 1% FAR are reported, along with Rank-1 accuracy (%).

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>0.1% FAR</th>
<th>1% FAR</th>
<th>Rank-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSLDA</td>
<td>13.81</td>
<td>36.38</td>
<td>47.92</td>
</tr>
<tr>
<td>RSLDA+Whitening</td>
<td>20.57</td>
<td>45.10</td>
<td>55.36</td>
</tr>
<tr>
<td>RSLDA+Whitening+HFA</td>
<td>42.42</td>
<td>70.13</td>
<td>71.43</td>
</tr>
</tbody>
</table>

Table 6.10: Evaluation of different face matchers on PCSO thermal dataset with HPOG descriptor. GAR (%) performances at both 0.1% FAR and 1% FAR are reported, along with Rank-1 accuracy (%).

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>0.1% FAR</th>
<th>1% FAR</th>
<th>Rank-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSLDA</td>
<td>21.19</td>
<td>44.16</td>
<td>57.14</td>
</tr>
<tr>
<td>RSLDA+Whitening</td>
<td>29.14</td>
<td>55.70</td>
<td>63.39</td>
</tr>
<tr>
<td>RSLDA+Whitening+HFA</td>
<td>47.92</td>
<td>74.32</td>
<td>74.40</td>
</tr>
</tbody>
</table>

better performance. We have also plotted the ROC and CMC curves for different face matchers in Figure 6.18.

**Image Filters**

Next, we provide a more detailed analysis on the use of different images filters with the RSLDA+Whitening+HFA matcher (PSIFT). The purpose is to show that each image filter provides discriminative information and their fusion should result in better performance. As seen from Table 6.11, CSDN image filter obtains GARs of 34.14% and 59.60% at 0.1% FAR and 1% FAR, respectively. The corresponding Rank-1 accuracy is 56.66%. The SQI image filter achieves very close performance, with reported GARs of 32.10% and 59.45% at 0.1% FAR and 1% FAR, respectively. On the other hand, the application of GIST image filter results in GARs of 22.11% and 49.71% at 0.1% FAR and 1% FAR, respectively. In summary, CSDN and SQI obtain better performances than GIST filter for this experiment. For individual image filters, they have demonstrated discriminative performances. Hence, the matching scores from individual image filters are fused and it results in increased performance.
Figure 6.18: ROC and CMC curves of the proposed matching method on the PCSO dataset for PSIFT and HPOG descriptors

Table 6.11: Evaluation of the proposed RSLDA+Whitening+HFA (PSIFT) method on PCSO thermal dataset with different image filters. GAR (%) performances at both 0.1% FAR and 1% FAR are reported, along with Rank-1 accuracy (%).

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>0.1% FAR</th>
<th>1% FAR</th>
<th>Rank-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDN</td>
<td>34.14</td>
<td>59.60</td>
<td>56.55</td>
</tr>
<tr>
<td>GIST</td>
<td>22.11</td>
<td>49.71</td>
<td>40.48</td>
</tr>
<tr>
<td>SQI</td>
<td>32.10</td>
<td>59.45</td>
<td>54.76</td>
</tr>
<tr>
<td>Fusion</td>
<td><strong>42.42</strong></td>
<td><strong>70.13</strong></td>
<td><strong>71.43</strong></td>
</tr>
</tbody>
</table>
Table 6.12: Comparison of the proposed matcher against state-of-the-art matchers on PCSO thermal dataset. The proposed method is the score-level fusion of PSIFT and PHOG with RSLDA+Whitening+HFA. GAR (%) performances at both 0.1% FAR and 1% FAR are reported, along with Rank-1 accuracy (%).

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>0.1% FAR</th>
<th>1% FAR</th>
<th>Rank-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>54.15</td>
<td>80.91</td>
<td>84.52</td>
</tr>
<tr>
<td>P-RS+D-RS [65]</td>
<td>72.7 ± 13.47</td>
<td>78.2 ± 0.13</td>
<td>49.2 ± 1.90</td>
</tr>
<tr>
<td>COTS [65]</td>
<td>44.4 ± 7.85</td>
<td>47.5 ± 2.49</td>
<td>21.5 ± 0.83</td>
</tr>
<tr>
<td>RS-PLS</td>
<td>22.93</td>
<td>50.30</td>
<td>58.63</td>
</tr>
<tr>
<td>RS-CCA</td>
<td>40.60</td>
<td>70.33</td>
<td>70.83</td>
</tr>
</tbody>
</table>

Comparison Against State-of-the-Art

We also compare against the performances of state-of-the-art THM to VIS facial matchers presented in [65]. Realizing that PSIFT and HPOG can provide complementary information, a simple sum-rule is used to fuse the matching scores. As can be seen from Table 6.12, the proposed method obtains GARs of 54.15% and 80.91% at 0.1% FAR and 1% FAR, respectively. Compared to P-RS+D-RS method, it obtains higher GAR for 1% FAR. Moreover, the reported Rank-1 accuracy of 84.52% is significantly higher than the reported 49.2 ± 1.90 accuracy for P-RS+D-RS. In forensic applications, the success of a face recognition system is measured by the number of correctly retrieved identities in matching THM probe images against a background dataset consisting of VIS images. In this sense, higher Rank-1 accuracy means more identities are correctly retrieved. This indicates the developed matcher is more suitable in identity retrieval applications.

Further, the proposed method is significantly better than the commercial matcher (COTS), RS-PLS and RS-CCA methods. Interestingly, RS-CCA matcher achieves better performance than RS-PLS. This suggests that the combination of CCA and random subspace is more effective than its counterpart. Regarding the computational efficiency, cascaded subspace learning takes approximately 4.74 seconds to build each random subspace. And CCA and PLS take approximately 1.53 and 10.98 seconds to build each random subspace.

---

6The listed results of D-RS+P-RS and COTS are directly cited from authors’ paper.
6.3.5 Identity Component Reconstruction

In Section 6.3.4, we have shown that the addition of HFA layer leads to improved matching performance. To better understand the role of HFA, we extract identity components from both VIS and THM images. In Eqn. (6.39), $Ux$ represents the identity component and $\beta$ is the mean vector of the population. Here, we use $Ux + \beta$ to reconstruct spectral-invariant images. The process for extracting spectral-invariant identity experiments is given below:

1. Both VIS and THM images are divided into overlapping patches\(^7\). Patches from the same position are used to form a patch-specific training dataset. HFA model is trained on each patch-specific training dataset.

2. Patches located in the same positions are sampled from an image in the test set and the corresponding identity components $Ux$ are extracted after obtaining $x$ from Eqn. (6.42).

3. The extracted identity components are used to reconstruct the patches and form the original image.

Specifically, patches of size $16 \times 25$ are sampled with a stride of 4 pixels in the horizontal direction and 10 pixels in the vertical direction. After the patches are reconstructed, pixels within the overlapped region are averaged. The identity preserving results are shown in Figure 6.19. According to the visualization results, the extracted identity components indicate that identity information is well reconstructed across spectra. It is also interesting to notice that the recovered identity component from thermal spectrum has clearly visible periocular regions. To verify the effectiveness of HFA model, we run an additional experiment where only HFA layer is used during the ensemble learning. The whitening transformation and LDA layers are not utilized. The reported GARs are 7.784% and 18.85% for 0.1% FAR and 1% FAR, respectively. The associated Rank-1 accuracy is 24.70%.

\(^7\)No image filters are utilized for this process
Figure 6.19: Illustration of generating spectral-invariant identity images across different spectra on PCSO test set. VIS and THM samples from the same subject are used here. These results are based on the learning of HFA model on PCSO training dataset. For each subject, top row shows the original VIS and THM images and bottom row shows the corresponding reconstructed spectral-invariant identity images.

### 6.3.6 Evaluation on Carl Dataset

In section 6.3.4, we have analyzed the performance of proposed face matchers on a single-database test. Cross-database experiment, on the other hand, can further evaluate if the matcher is capable of generalizing well. Carl Database [165] contains 41 subjects that are simultaneously acquired in visible, near-infrared and thermal spectra. Here, 5 samples per subject from visible spectrum are used in the gallery, and 5 samples per subject from thermal spectrum are used as probes. Some of the subjects are shown in Figure 6.20. The purpose of this experiment is to test if the matcher trained using the PCSO dataset generalizes well to an unseen dataset. In other words, all the matchers are trained exclusively on the PCSO dataset and the ensuing matchers are used to perform the face matching between VIS and THM in Carl dataset.

As shown in Table 6.13, RSLDA+Whitening+HFA (PSIFT) obtains the best performance among all the matchers. It obtains GARs of 18.24% and 38.63% at 0.1% FAR and 1% FAR, respectively. The corresponding Rank-1 accuracy is 56.10%. For the HPOG descriptor in Table 6.14, RSLDA+Whitening+HFA obtains GARs of 29.03% and 48.88% at 0.1% FAR and 1% FAR, respectively. Next, we plot the ROC and CMC curves for all the
Figure 6.20: Samples of face images in VIS (top) and THM (bottom) spectra. Images are from Carl database and not co-registered.

Table 6.13: Evaluation of different face matchers on Carl thermal dataset with PSIFT descriptor. GAR (%) performances at both 0.1% FAR and 1% FAR are reported, along with Rank-1 accuracy (%).

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>0.1% FAR</th>
<th>1% FAR</th>
<th>Rank-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSLDA</td>
<td>4.232</td>
<td>14.27</td>
<td>41.95</td>
</tr>
<tr>
<td>RSLDA+Whitening</td>
<td>5.357</td>
<td>18.15</td>
<td>49.76</td>
</tr>
<tr>
<td>RSLDA+Whitening+HFA</td>
<td>18.24</td>
<td>38.63</td>
<td>56.10</td>
</tr>
</tbody>
</table>

matchers in Figure 6.21. In Table 6.15, we also compare the proposed method against the baseline algorithms. We have observed decreasing performances when the proposed matcher is trained on PCSO dataset and tested on the Carl dataset. Nevertheless, such a phenomenon is commonly perceived in face recognition when a matcher trained on one database is used to test on another database. Moreover, sensor differences of the thermal cameras between PCSO and Carl datasets is also a contributing factor for the performance decrease. Considering the challenging nature of face matching between VIS and THM, the achieved performances are still very promising for the proposed matcher.

On the other hand, RS-CCA obtains GARs of 10.56% and 21.22% at 0.1% FAR and 1% FAR, respectively. RS-PLS obtains lower GAR performances. In this regard, it is evident that the proposed method performs much better in cross-database test. This demonstrates that PLS and CCA methods tend to overfit the training set since their performances drop significantly compared to single-database test.
Figure 6.21: ROC and CMC curves of different face matchers on Carl dataset for PSIFT and HPOG descriptors.
Table 6.14: Evaluation of different face matchers on Carl thermal dataset with HPOG descriptor. GAR (%) performances at both 0.1% FAR and 1% FAR are reported, along with Rank-1 accuracy (%).

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>0.1% FAR</th>
<th>1% FAR</th>
<th>Rank-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSLDA</td>
<td>9.05</td>
<td>20.22</td>
<td>50.24</td>
</tr>
<tr>
<td>RSLDA+Whitening</td>
<td>14.29</td>
<td>30.02</td>
<td>69.27</td>
</tr>
<tr>
<td>RSLDA+Whitening+HFA</td>
<td>29.03</td>
<td>48.88</td>
<td>70.24</td>
</tr>
</tbody>
</table>

Table 6.15: Comparison of the proposed matcher against state-of-the-art matchers on Carl thermal dataset. The proposed method is the score-level fusion of PSIFT and PHOG with RSLDA+Whitening+HFA. GAR (%) performances at both 0.1% FAR and 1% FAR are reported, along with Rank-1 accuracy (%).

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>0.1% FAR</th>
<th>1% FAR</th>
<th>Rank-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>27.71</td>
<td>51.24</td>
<td>75.61</td>
</tr>
<tr>
<td>RS-PLS</td>
<td>3.42</td>
<td>12.64</td>
<td>23.90</td>
</tr>
<tr>
<td>RS-CCA</td>
<td>10.56</td>
<td>21.22</td>
<td>32.20</td>
</tr>
</tbody>
</table>

6.4 Summary

In this chapter, we studied the problem of coding Gabor filter responses for performing effective face recognition. Our proposed coding method, referred to as Local Gradient Gabor Pattern (LGGP), is based on utilizing gradient information on Gabor-transformed images and was demonstrated to be effective compared to LBP-like coding schemes when encountering face images with variations in expression and illumination. Furthermore, we presented the proposed coding method as a potential solution to the cross-spectral face recognition problem of matching near-infrared images against visible spectra images. Finally, we established the utility of the proposed coding scheme in soft biometric applications. A more fundamental analysis is required to understand the photometric and structural variations in images across spectral bands. The descriptor may have to be regularized in order to account for the dynamics of these variations. Next, we presented a method for matching long-distance NIR with VIS face images. Experiments have demonstrated that the proposed method is effective in handling cross-spectral and cross-distance face matching, outperforming previ-
ous existing methods by a large margin. The proposed method uses a pyramid-based SIFT representation which captures information across distance, along with three different image filters. RS-LDA method is further utilized to learn the discriminant projections, followed by sparse representation classifier. The final output is the fusion of matching scores from all image filters. To further reduce the cross-spectral matching impact, whitening transformation is utilized to learn a more discriminant subspace. Experimental results on the LDHF database demonstrate the effectiveness of the proposed method.

Finally, we proposed a method for matching thermal images against visible images. We provide extensive evidence that the proposed method is effective in handling this problem and is comparable to state-of-the-art HFR matchers. The proposed method generates multiple subspaces by a cascaded subspace learning based on whitening transformation, factor analysis and discriminant analysis models. The final output is the fusion of matching scores from individual image filters. Both single-dataset and cross-dataset tests demonstrate the effectiveness of the proposed method.
Chapter 7

Summary and Future Work

7.1 Summary

This dissertation addressed two challenging problems in heterogeneous face recognition: facial cosmetics and multispectral imagery. The primary contribution of this dissertation is the design of a novel patch-based ensemble learning method in conjunction with a cascaded subspace learning process to perform effective heterogeneous face recognition.

In Chapter 2, we identify the impact of makeup on state-of-the-art face recognition matchers and offer the following observations:

- Existing face recognition matchers are vulnerable to the application of facial makeup.
- The impact of facial makeup is observed across different ethnicity groups.

Chapter 3 extends the makeup impact analysis to automated gender and age estimation algorithms and offers the following observations:

- Gender spoofing via the application of makeup is possible.
- Makeup has a major impact on age estimation algorithms and, therefore, developing makeup-robust estimation algorithms deserve immediate attention.

Chapter 4 proposes a solution to detect the presence or absence of makeup and offers the following observations:
Development of robust makeup detection algorithms are beneficial when performing face recognition across cosmetics.

Such a makeup detection algorithm can be potentially used in addressing other challenges encountered by automated gender and age estimation algorithms.

Chapter 5 presents a novel makeup-robust face recognition algorithm and offers the following observations:

- Developing feature representation schemes that are robust to cosmetic changes are crucial to the generation of successful matchers.
- Learning multiple common discriminant subspaces between before-makeup and after-makeup images (patches) is the key to minimize makeup variations.
- The need for using a training set that has images with makeup variation is critical in designing a makeup-robust face recognition matcher.

Finally in Chapter 6, we design several matching algorithms to address cross-spectral face recognition and provide the following observations:

- Matching remote NIR face images against VIS face images is still particularly challenging due to large variations in the case of cross-distance face images.
- Designing effective image filters to facilitate cross-spectral face matching and utilizing multiresolution image representation schemes are two simultaneous factors that need to be carefully considered in remote heterogeneous face recognition.
- A common discriminant subspace between VIS and THM cannot be learned effectively by using a single type of subspace transformation due to the large appearance difference; therefore a cascaded subspace learning scheme based on whitening transformation, factor analysis model and common discriminant analysis model is utilized.
- A thermal face image can be decomposed into two components: an identity component that is supposed to be reasonably invariant to spectral variation and a spectral component that reflects the sensor variation.
7.2 Future Work

Though this dissertation addresses the challenges of face recognition across facial cosmetics and multispectral imagery, several questions remain to be answered.

The proposed solution of matching after-makeup images to before-makeup images has shown promising results. However, the impact of makeup is still noticeable. Some of the ideas that stem from the work of cross-spectral face recognition can be utilized to further improve the matching performance. For instance, whitening transformation and factor analysis model can be used to construct robust subspaces within the SRS-LDA framework.

The proposed framework for matching remote NIR images to VIS images has demonstrated the importance of accounting for image degradations during the matching process. In this dissertation, we simulate some of the degradations, such as blur and low resolution, in the training set. However, the noise factor cannot be simulated in the training dataset because we do not have prior knowledge about the sensor noise. Therefore, a training dataset that reflects the sensor conditions should help further improve the matching performance. Additionally, some image restoration methods can be harnessed to reduce the degradations.

Inspired by the use of factor analysis model for face recognition across pose and age variations, we successfully apply the model to tackle face matching between VIS and THM images. Our analysis shows that identity components can be recovered across different spectra. Nevertheless, direct matching of the recovered identity components across spectra delivers poor performance. This is due to the artifacts introduced by patch-to-patch image synthesis. More advanced image synthesis methods (e.g., Poisson editing) can be incorporated to provide better visualization for recognition purposes.

Another emerging and interesting topic is the potential of using makeup for spoofing an identity, where an individual attempts to look like another person. In this regard, we need to define spoofing indices that quantify the potential of makeup for spoofing identities. Further, we would also like to answer the following questions: (a) Is a poor matcher less vulnerable to makeup-induced identity spoofing? and (b) Is a good matcher more vulnerable to makeup-induced identity spoofing? Answering these two related questions will advance our understanding of not only makeup-induced identity spoofing, but also other identity
spoofing attacks. We are also interested in the problem of determining the degree of makeup applied to the face - this will have benefits in obfuscation/spoofing scenarios.
Chapter 8

Appendix

8.1 List Of Publications


- C. Chen and A. Ross, “Local Gradient Gabor Pattern (LGGP) with Applications in Face Recognition, Cross-spectral Matching and Soft Biometrics,” SPIE Biometric and
Surveillance Technology for Human and Activity Identification X, (Baltimore, USA), May 2013.


References


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


[144] Sifei Liu, Dong Yi, Zhen Lei, and Stan Z. Li, “Heterogeneous face image matching using multi-scale features,” in International Conference on Biometrics, 2012, pp. 79–84.


REFERENCES


